

**IMPLEMENTACIÓN DE UNA HERRAMIENTA DE RADIO DEFINIDO POR
SOFTWARE QUE EMULE UNA RED DE TELEFONÍA CELULAR MÓVIL PARA
SER USADA EN LA ENSEÑANZA DENTRO DEL PROGRAMA DE INGENIERÍA
ELECTRÓNICA**

**ARTURO JOSÉ OSPINO POLANCO
CAMILO ANDRES DIAZ VILLADIEGO**



**UNIVERSIDAD DE LA COSTA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y ELECTRÓNICA
BARRANQUILLA**

2020

**IMPLEMENTACIÓN DE UNA HERRAMIENTA DE RADIO DEFINIDO POR
SOFTWARE QUE EMULE UNA RED DE TELEFONÍA CELULAR MÓVIL PARA
SER USADA EN LA ENSEÑANZA DENTRO DEL PROGRAMA DE INGENIERÍA
ELECTRÓNICA**

**ARTURO JOSÉ OSPINO POLANCO
CAMILO ANDRES DIAZ VILLADIEGO**

**PROYECTO DE GRADO PARA OBTENER EL TÍTULO DE
INGENIERO ELECTRÓNICO**

TUTOR

MsC. GABRIEL DARIO PIÑERES ESPITIA

COTUTOR

PhD. ALEJANDRO CAMA PINTO

UNIVERSIDAD DE LA COSTA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y ELECTRÓNICA

BARRANQUILLA

2020

Dedicatoria

ARTURO JOSÉ OSPINO POLANCO: Dedico este trabajo principalmente a mi padre Arturo Ospino Vargas y Darcy Polanco Vargas porque me dieron el apoyo económico para terminar esta carrera.

Dedico, ante todo, a Dios por haberme dado la sabiduría, la constancia y el entendimiento.

A mis amistades por ese apoyo incondicional, sus consejos y serenidad. Amigos, conocidos y familiares entre ellos, Kevin acuña Payares, Cecilia Gonzales Acuña, y otros que no he mencionado pero que llevo en mi corazón, gracias.

CAMILO ANDRES DIAZ VILLADIEGO: Dedico este proyecto de grado a mis padres Carlos Diaz y Consuelo Villadiego, a mi tía Leslie Villadiego y abuelos, por todo el apoyo que he recibido de su parte desde el inicio de mi profesión, con esta mención quiero expresar mi gratitud.

Agradecimientos

Agradecemos a Gabriel Piñeres por su guía y conocimientos que han ayudado a completar el desarrollo del proyecto también a Cama Pinto por su gran sabiduría y se acompañamiento en los temas fundamentales y gran apoyo en el desarrollo del proyecto.

Agradecemos el apoyo de la Universidad De La Costa y al programa de ingeniería electrónica, por el apoyo en el préstamo de las herramientas y espacios necesarios para llevar a cabo el proyecto; también a la ingeniera Zhoe Comas por su apoyo a lo largo del proyecto.

Resumen

La telefonía celular es una de las formas más básicas de comunicación en la actualidad, la cual requiere de un entendimiento por parte de los profesionales que se forman en este campo. Con el fin de ampliar el acceso a esta tecnología se desarrolla una herramienta de radio definido por software que permita analizar el funcionamiento de un sistema de telefonía celular GSM; para esto se realizó una revisión del estado actual de la literatura bajo 3 ejes temáticos, Tendencias en Radio Definido por Software, Tendencias en Sistemas basados en OpenBTS y Tendencias en herramientas para modelamiento e implementación de un sistema de telefonía celular móvil. Para llevar a cabo este trabajo se utilizó el dispositivo USRP N210, el software libre OpenBTS y Asterisk en el sistema operativo Ubuntu versión 12.04, el estándar de comunicación que se usó el GSM, con el fin de implementar un sistema de radio definido por software. El desarrollo del trabajo se dividió en 4 fases; la primera identificó las características del prototipo de OpenBTS a desarrollar, en la que se determinaron los componentes necesarios para implementar el sistema y se buscaron precedentes; la segunda permitió la caracterización del sistema a desarrollar, definiéndose las herramientas de software y hardware; en la tercera se desarrolló la implementación y configuración del sistema propuesto; finalmente en la última fase se realizaron pruebas en el sistema. En base a lo anterior se realizaron distintas pruebas con el fin de determinar, máximo alcance de la señal en espacio libre y urbano; nivel de pérdida en las llamadas; conexión de múltiples teléfonos móviles; calidad de llamada; envío de mensajes de texto. El resultado de las pruebas estableció que el dispositivo tiene un alcance de 120 m en espacio libre, su calidad de llamada y envío de mensajes son buenos.

Palabras clave: OpenBTS, Radio definido por software, Sistema de Telefonía celular móvil GSM, Asterisk, USRP N210

Abstract

Cellular telephony is one of the most basic forms of communication today, which requires an understanding by professionals trained in this field. In order to expand the access to this technology, a software defined radio tool was developed to analyze the operation of a GSM cellular telephony system; for this purpose, a review of the current state of the literature was carried out under 3 thematic axes, Trends in Software Defined Radio, Trends in Systems based on OpenBTS and Trends in tools for the modeling and implementation of a mobile cellular telephony system. To carry out this work we used the USRP N210 device, the free software OpenBTS and Asterisk in the operating system Ubuntu version 12.04, the communication standard used by the GSM, in order to implement a software defined radio system. The development of the work was divided in 4 phases; the first one identified the characteristics of the OpenBTS prototype to be developed, in which the necessary components to implement the system were determined and precedents were looked for; the second one allowed the characterization of the system to be developed, defining the software and hardware tools; in the third one the implementation and configuration of the proposed system was developed; finally in the last phase tests were made in the system. Based on the above, different tests were carried out in order to determine the maximum range of the signal in free and urban space; level of loss in the calls; connection of multiple mobile phones; call quality; sending of text messages. The result of the tests established that the device has a range of 120 m, its call quality and sending of messages are good.

Keywords: OpenBTS, Software Defined Radio, GSM Mobile Phone System, Asterisk, USRP N210

Contenido

Lista de tablas y figuras.....	4
Introducción	12
1. Descripción del Problema	15
2. Justificación.....	16
3. Objetivos de la Investigación	17
3.1. Objetivo General	17
3.2. Objetivos Específicos.....	17
4. Marco Teórico y Estado Del Arte	17
4.1. Conceptos y Definiciones	17
4.1.1. Sistema GSM.....	17
4.1.2. OpenBTS	20
4.1.3. Asterisk.....	20
4.1.4. Ubuntu	21
4.1.5. Radio Definido por Software (SDR).....	22
4.1.6. USRP (Universal Software Radio Peripheral)	22
4.2. Revisión del estado actual de la literatura.....	24
4.2.1 Tendencias en Radio Definido por Software	24
4.2.2 Tendencias en Sistemas basados en OpenBTS	28
4.2.3 Tendencias en herramientas para modelamiento e implementación de un sistema de telefonía celular móvil.....	32
5. Metodología	35

5.1.	Tipo de Estudio	35
5.2.	Identificación y caracterización de la herramienta OpenBTS a desarrollar.	41
5.3.	Implementación de la OpenBTS para el Programa de Ingeniería Electrónica de la Universidad de la Costa.	51
5.3.1.	Configuración de la herramienta de radio definido por software para el envío de mensajes SMS y llamadas de voz GSM.....	51
5.3.2.	Configuración y ejecución del OpenBTS en el radio USRP.....	63
5.3.3.	Configuración de un sistema de conmutación basado en Asterisk para el OpenBTS.	66
6.	Resultados y Análisis	82
6.1.	Análisis de la capacidad de llamadas y mensajes según el número de equipos conectados.....	83
6.2.	Análisis del consumo de recursos por parte de la herramienta OpenBTS	84
6.2.1.	Calidad de llamadas y mensajes SMS según la distancia cubierta.	86
7.	Conclusiones	90
7.1	Trabajos futuros.....	90
8.	Referencias	92

Lista de Tablas y Figuras

Tablas

Tabla 1 Descripción del desarrollo de las fases del proyecto.....	35
Tabla 2 Especificaciones del equipo de cómputo que usa openbts.....	50
Tabla 3 Lista de dependencias para openbts.....	51
Tabla 4 Librerías adicionales para Asterisk.....	67
Tabla 5 Equipos compatibles y usados en la prueba de conectividad de usuarios.....	75
Tabla 6 Datos de transferencias de datos.....	78
Tabla 7 Datos de procesamiento de consumo del computador.....	79
Tabla 8 Mediciones de nivel de señal.....	81
Tabla 9 Mediciones de nivel de señal.....	82

Figuras

Figura 1 Autores más relevantes en la temática de Radio definido por software.....	25
Figura 2 Cantidad de publicaciones por países en la temática de Radio definido por software.....	25
Figura 3 Documentos publicados por año en la temática de Radio definido por software (fuente propia).	26

El análisis de este eje temático utilizó la cadena de búsqueda (OpenBTS AND systems), con la herramienta scopus; conforme a esta búsqueda se tiene que en la figura 4 se ven los “autores más importantes”, en la figura 5, se aprecian los países donde más trabajos se ha investigado esta temática y en la figura 6. se identifican los trabajos arrojados por la búsqueda desde el año 2010 hasta enero del 2019.

Documents by author

Compare the document counts for up to 15 authors.

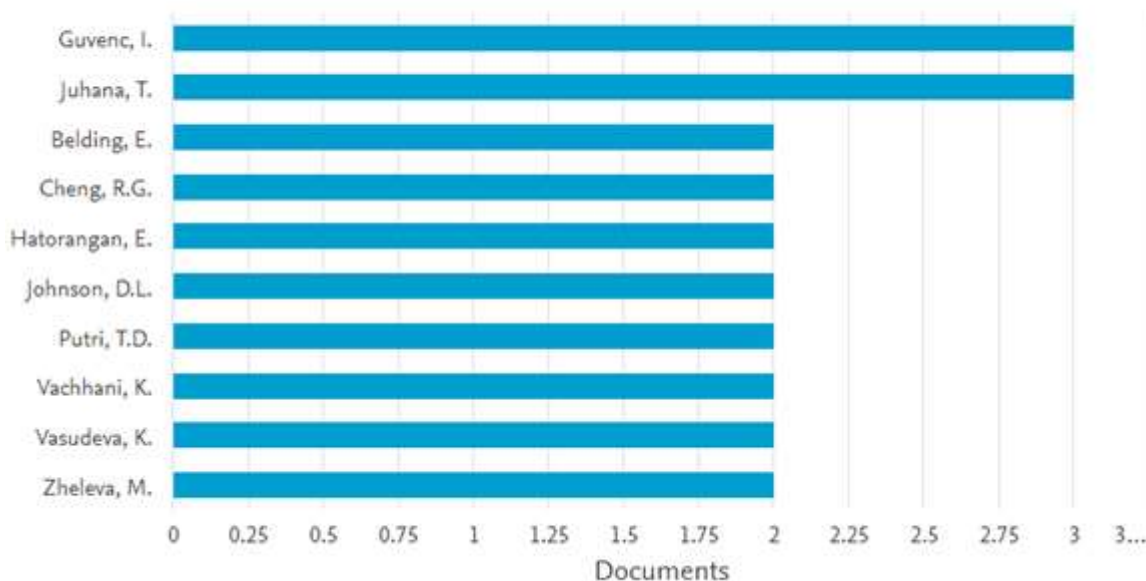


Figura 4 Autores más importantes en la temática de OpenBTS (fuente propia). 28

Figura 5 Países con más publicaciones en la temática de OpenBTS (fuente propia). 29

Figura 6 Documentos por años en la temática de OpenBTS (fuente propia). 30

Figura 7 Autores más relevantes en la temática de modelamiento y/o implementación de sistemas de telefonía celular móvil (fuente propia). 32

Figura 8 Publicaciones por países en la temática de modelamiento y/o implementación de sistemas de telefonía celular móvil (fuente propia). 33

Figura 9 Publicaciones por año en la temática de modelamiento y/o implementación de sistemas de telefonía celular móvil (fuente propia). 33

Figura 10 Comparación del sistema OpenBTS con un sistema GSM tradicional (Manuel Quiñones, 2015). 42

Figura 11 Esquema de operación del sistema OpenBTS (Manuel Quiñones, 2015). 43

Figura 12 Especificaciones de la USRP N210 (Ettus, 2020). 46

Figura 13 Ettus N210 parte interna (fuente propia). 47

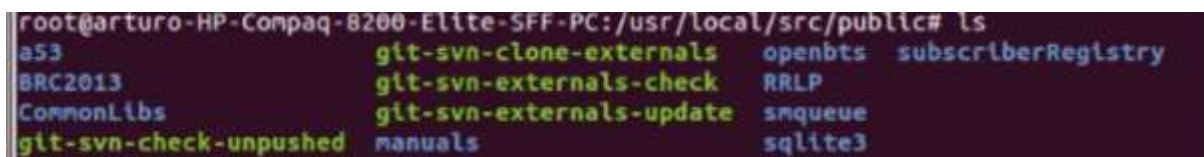
Figura 14 WBX 50-2200 MHz Rx/Tx (40 MHz) (fuente propia). 47

Figura 15 Ettus N210 armado y conexión (fuente propia). 48

Figura 16 Diagrama de bloques para describir el funcionamiento del UHD de una USRP (Ettus, 2020) 49

Figura 17 Archivo. bashrc (fuente propia). 59

Figura 18 Respuesta del comando uhd_find_devices sin la USRP conectada (fuente propia). 60



```
root@arturo-HP-Compaq-8200-Elite-SFF-PC:/usr/local/src/public# ls
a53          git-svn-clone-externals  openbts  subscriberRegistry
BRC2013      git-svn-externals-check  RRLP
CommonLibs   git-svn-externals-update snqueue
git-svn-check-unpushed  manuals                sqlite3
```

Figura 19 Visualización de la carpeta public (fuente propia). 61

Figura 20 Menú de configuración de red cableada (fuente propia). 64

Figura 21 Menú de ajustes instalación Asterisk (fuente propia). 68

Figura 22 Equipos con llamadas en curso usando herramienta OpenBTS (fuente propia).

76

Figura 23 Ubicación de la herramienta OpenBTS para la prueba 2 (fuente propia). 77

Figura 24 Monitor de sistema de Ubuntu que muestra consumo de recursos al usar OpenBTS (fuente propia). 78

Figura 25 Ubicación de la herramienta OpenBTS para los test 1 y 2 (fuente propia). 81

Figura 26 Gráficos de llamadas y mensajes en simultáneo (fuente propia). 84

Figura 27 Comparativa de consumo de recursos ejecutando la herramienta OpenBTS y sin utilizar este equipo (Fuente propia). 85

Figura 28 Gráfica de niveles de RSSI obtenidos en el equipo receptor con la herramienta OpenBTS ubicada en el piso 3 del bloque 11 de la Universidad de la Costa Nivel de RSSI (dBm) vs Distancia (m) (Fuente propia). 87

Figura 29 Gráfica de niveles de RSSI obtenidos en el equipo receptor con la herramienta OpenBTS ubicada en el piso 7 del bloque 11 de la Universidad de la Costa Nivel de RSSI (dBm) vs Distancia (m) (Fuente propia). 88

Introducción

Desde sus inicios el ser humano ha tenido la necesidad innata de comunicarse y ha desarrollado varios métodos con el fin solventar esta necesidad; en la actualidad la forma más común de comunicarnos a grandes distancias es usando nuestro teléfono móvil, conectando este a una red, generalmente instaurada por una compañía de telefonía móvil. Sin embargo, es complejo instaurar radiobases de telefonía celular; una solución que pueda satisfacer esta necesidad de manera abierta y libre es el radio definido por software (SDR). Esta es una herramienta que permite realizar todos los procesos de un hardware en especial y que permite en casos puntuales simular la operación de radiobases de telefonía celular. A través de ella se realizan tareas tales como, modulación, amplificación, etc, a través de software, dejando de lado el hardware específico (EcuRed contributors, 2014), lo que la convierte en una herramienta muy útil en la actualidad, ya que permite el acceso a la tecnología de comunicación a teléfonos móviles. Esto genera un beneficio al público general, dándole la oportunidad a estudiantes, investigadores, emprendedores, comunidades remotas y demás personas interesadas a desarrollar y aprovechar el acceso a la comunicación (Instituto de Ingeniería Eléctrica, 2016).

Es por esto que dentro de las principales aplicaciones de las herramientas de radio definido por software se tienen: la industria privada, donde se pueden usar con el fin de la comunicación interna dentro de una empresa por ejemplo una red privada (universidad de vigo, 2018); además, se ha realizado la implementación de servicios de telefonía celular en zonas remotas, a través de sistemas OpenBTS (Prasannan, y otros, 2013); otro caso de OpenBTS es en situaciones de cataclismos, cuando las redes de celular están fuera de servicio

debido a catástrofes naturales (Kinjal Aggrawal, 2017). Se puede dar fe del impacto de los sistemas basados en OpenBTS, tomando ejemplo el proyecto realizado por un grupo de ingenieros veteranos que lograron instalar un sistema GSM en una comunidad con unos costes de instalación bajos, de aproximadamente 4,500 dólares, cobrando a sus clientes 2 dólares por la prestación del servicio mensual; esto demuestra una mejora sustancial en costes, en comparación con las implementaciones tradicionales (GOODIER, 2015).

Conforme a lo anterior, en este trabajo se realizó el desarrollo de una herramienta de radio definido por software con el fin de implementar un sistema GSM en el Programa de Ingeniería Electrónica de la Universidad de la Costa. Este trabajo desarrolla un prototipo de OpenBTS utilizando un radio USRP (Universal Software Radio Peripheral) N210 de la empresa ETTUS, en donde se plantea un diseño inicial, se detalla la implementación de la herramienta y se realiza una metodología de pruebas; finalmente se realizan pruebas de funcionamiento de la herramienta, teniéndose que el radio alcanza una distancia de 120 metros de cobertura a una altura de 19 metros, en una zona despejada, sin obstáculos; también el consumo de procesamiento, la calidad de llamadas son evaluadas, y otras pruebas consideradas relevantes que permiten entender el funcionamiento de la herramienta.

Este documento está organizado de la siguiente manera: posterior a la introducción se encontrará la descripción del problema a resolver, la justificación y los objetivos planteados, para completar el capítulo 1. El siguiente capítulo detalla los principales aspectos del marco teórico, describiendo los principales ejes temáticos sobre los que se exponen trabajos similares al propuesto en este proyecto. El capítulo 3 se enfoca en describir la metodología utilizada, en donde se plantearon cuatro fases, partiendo de la identificación de la herramienta OpenBTS a desarrollar; posteriormente, se caracterizó la herramienta OpenBTS propuesta en cuanto a sus partes y componentes; seguidamente se implementó la herramienta OpenBTS,

realizando las conexiones y configuraciones requeridas; finalmente se realizaron pruebas de envío de mensajes y llamadas, verificando la operación de la herramienta OpenBTS. El capítulo 4 describe los principales resultados de este proyecto, en donde se destacan el análisis de las pruebas realizadas con el fin de verificar el funcionamiento de la herramienta OpenBTS. Finalmente, las principales conclusiones son expuestas.

1. Descripción del Problema

Uno de los campos de estudio que abarca la Ingeniería Electrónica es la telefonía celular móvil, una tecnología de amplio uso en telecomunicaciones que es de importante aplicación en el mundo actual. Su estudio y aprendizaje requiere herramientas que permitan a los estudiantes comprender su arquitectura, operación y servicios.

Actualmente un problema que se presenta en el programa de Ingeniería Electrónica de Universidad de la Costa es no contar con herramientas de hardware que permitan el estudio de las redes celulares GSM. Esto va asociado a causas como el costo de los equipos y licencias que usan estos sistemas. En otras palabras, la poca existencia de equipos y herramientas que permitan el aprendizaje de la telefonía celular es una limitante en el aprendizaje actual de los sistemas GSM.

Herramientas como OpenBTS (openbts.org, 2012), Asterisk (quarea, 2015), wireshark ([wireshark](http://wireshark.org), 2020) y FreePBX (3cx, 2020) permiten el estudio de la telefonía celular móvil, pero dentro del Programa de Ingeniería Electrónica no se cuentan con estas en la actualidad; este tipo de herramientas serían de gran aporte en distintas asignaturas, como por ejemplo, la electiva de profundización “Telefonía Celular Móvil”; además asignaturas del ciclo profesional como “Propagación y medios de transmisión”, y nuevas asignaturas en el área de Sistemas de Telecomunicaciones y posgrados serían beneficiadas de contar con estas herramientas.

Con base en la anterior problemática planteamos el siguiente cuestionamiento:

¿Cómo desde la temática de Radio Definido por Software se puede mejorar el aprendizaje de la temática de Redes de telefonía celular móvil GSM en el área de sistemas de telecomunicaciones del programa de Ingeniería Electrónica de la Universidad de la Costa?

Sin duda el aprendizaje de esta temática es de actualidad, ya que en nuestro medio local, regional y nacional se cuenta con empresas de este sector que demandan de personal cualificado con conocimientos en redes de telefonía celular móvil.

2. Justificación

Las redes de telefonía celular móvil son objeto de amplio estudio y profundización, por ello, es necesario contar con herramientas que posibiliten dicho fin. Actualmente en el programa de Ingeniería Electrónica de la Universidad de la Costa no se cuenta con sistemas basados en radio definido por software para emular sistemas de telefonía celular móvil. Herramientas de este tipo suelen ser costosas y de licencia propietaria.

Conforme a lo anterior, se busca implementar una herramienta de radio definido por software que emule una red de telefonía móvil celular GSM. Esta herramienta es importante en el estudio de las redes celulares y mejora la preparación de los estudiantes en este campo. La herramienta estará basada en el entorno OPENBTS (Burgess and Samra, 2008) ya que, al ejecutarlo en un equipo de bajo costo, como un radio USRP (radio definido por software) se tiene una estación de telefonía celular. Este sistema no posee la complejidad de los sistemas de telefonía móviles actuales; además, tiene una licencia GPL que no tiene las restricciones comerciales y privadas, lo que hace posible ser usada sin problema en el estudio e investigaciones de las redes de telefonía móvil celular.

La implementación de este sistema permitirá complementar el aprendizaje de las redes de telefonía móvil celular en programas de pregrado y posgrados que así lo requieran, y así promover investigaciones y aplicaciones que se puedan generar posterior a su desarrollo.

3. Objetivos de la Investigación

3.1. Objetivo General

Implementar una herramienta de radio definido por software que emule una red de telefonía móvil celular GSM para ser usada en la enseñanza dentro del Programa de Ingeniería Electrónica.

3.2. Objetivos Específicos

- Identificar las características principales de un sistema de telefonía móvil celular GSM.
- Caracterizar una herramienta de radio definido por software que permita emular un sistema de telefonía móvil celular GSM.
- Configurar la herramienta de radio definido por software para el envío de mensajes SMS y llamadas de voz GSM.
- Realizar pruebas de conectividad de la herramienta de usuarios móviles a través del envío de mensajes y llamadas de voz GSM.

4. Marco Teórico y Estado Del Arte

4.1. Conceptos y Definiciones

4.1.1. Sistema GSM

Este es un estándar descrito por sus siglas, las que en español significan (Sistema Global de Comunicaciones Móviles); fue implementado en el año 1982, realizando un cambio de paradigma, migrando de lo analógico a lo digital, y dando comienzo a la tecnología 2G (Universidad Internacional de Valencia, 2018).

Es un sistema de comunicación que funciona a través de ondas de radio, donde sus estaciones se encuentran interconectadas, disponiendo de un controlador que se encarga de optimizar los recursos de la radio base con el fin de brindar la mejor calidad posible; estos controladores se encuentran conectados a un centro de cómputo a través de cables e interfaces; su banda de frecuencia fue definida dependiendo el territorio en el que se utilizaba, en así que en Europa se utilizó el espectro radioeléctrico de 900 y 1800 MHZ, y en estados unidos de 1900 MHZ, lo que complicó la compatibilidad de los equipos móviles en los distintos continentes. Dentro de la arquitectura de un sistema GSM se evidencian diversos componentes los que son: Sistema móvil (SM,) Estación base transceptora (BTS), Controlador de estación base (BSC), Central de conmutación móvil (MSC), bases de datos de control o auxiliares, Registro de localización del visitante (VLR), Registro de identidad del equipo (EIR), Registro de localización base (HLR) y Centro de autenticación (AUC) (Universidad Internacional de Valencia , 2018); Estos conceptos se definen a continuación.

SM (Estación móvil): En los sistemas de radiotelefonía móvil es el equipo transportable, radioteléfono móvil o portátil, con el que se desplaza el usuario de la red y que, dependiendo del sistema, permite una mayor o menor movilidad (glosarios servidor-alicante, 2016).

BTS (Estación base transceptora): Una estación base es una instalación fija o moderada de radio. Su función primordial es proporcionar a la estación base del sistema de telefonía móvil, la radio transmisión y recepción. Puede contener uno o más transceptores que puedan suministrar la capacidad de canales requeridos. Un transceptor es un dispositivo que cuenta un transmisor y un receptor dentro de un mismo circuito o recipiente. La estación base sirve a modo de punto de acceso a una red de comunicación fija (ya sea internet o telefonía) o que dos terminales se comuniquen con la estación base. Una de sus finalidades es conectarse a radios de baja potencia, por ejemplo, equipos móviles (Roberto, 2015).

BSC (Controlador de estación base): Sirve a modo de un mediador por medio las estaciones base y los MSC, al tiempo que proporciona vías de voz a teléfonos móviles y otros dispositivos compatibles, un ejemplo podría ser una línea terrestre o Internet. Sus principales funciones son la administración de la red de radio (el control de radiofrecuencia y potencia), la administración del traspaso de llamadas de BTS (handover o traspaso de celdas) y la configuración de llamadas. La BSC diferencia el tráfico de voz y de datos ya que usan canales distintos, también actúa a modo de concentrador reduciendo el número de conexiones hacia el MSC (Roberto, 2015).

MSC (Central de conmutación móvil): es el encargado de tomar decisión sobre el enrutamiento de la llamada y ejecutar la conmutación correspondiente basándose en la información de la base de datos. Sus principales funciones son establecimiento de una llamada, enrutamiento de las llamadas, gestión de los abonados móviles (registro, autenticación, traspaso), terminación de cualquier llamada, control de servicios suplementarios y función de interconexión (conecta la red móvil a la red fija) (Roberto, 2015).

BASES DE DATOS DE CONTROL O AUXILIARES: de acuerdo con esta base de datos el MSC toma sus respectivas decisiones y hay de diferentes tipos. Las diferentes bases de datos de control, encontramos los siguientes elementos clave:

HLR (Registro de localización base): Se trata de la base de datos relacionada con el abonado y su información de localización (Roberto, 2015).

VLR (Registro de localización del visitante): Es la base de datos de los abonados de la zona (Roberto, 2015).

EIR (Registro de identidad del equipo): Almacena los datos de los equipos móviles. IMEI válidos e inválidos (Roberto, 2015).

AUC (Centro de autenticación): Se corresponde con la base de datos de los números secretos de autenticación contenidos en la SIM (Roberto, 2015)

4.1.2. OpenBTS

Es un método de acceso a una red móvil GSM basado en software libre, el que permite usuarios con teléfonos móviles compatibles con el estándar GSM, conectarse a una red independiente a la de los operadores tradicionales del mercado. El significado de sus siglas es Open Base Transceiver Station - Estación base de transceptor; debido a su popularidad entre los conocedores este software ha creado una comunidad de desarrollo, llamada openbts.org, la que se describe a sí misma a modo de “un proyecto de software de código abierto dedicado a revolucionar las redes móviles mediante la sustitución de protocolos de telecomunicaciones antiguos y sistemas de hardware patentados tradicionalmente con complejos protocolos de Internet y una arquitectura de software flexible. Esta arquitectura está abierta a la innovación por cualquiera, lo que permite el desarrollo de nuevas aplicaciones y servicios y simplifica drásticamente la configuración y operación de una red móvil” (openbts.org, 2012).

4.1.3. Asterisk

Es un programa con software libre que tiene la funcionalidad de una central telefónica o (PBX) la que significa central privada automática; este software está licenciado por GNU (licencia pública general), nació en el año 1999 y desde entonces ha venido evolucionando gracias a su comunidad de desarrolladores, permitiendo realizar las mismas funciones de una central telefónica, buzón de voz, conferencia, respuestas de voz interactiva, distribución

automática de llamadas y otras; cualquier persona tiene la capacidad de aportar a la comunidad, programado nuevas funciones en cualquier lenguaje compatible con GNU/Linux.

Esta herramienta de software permite conectarse a un determinado número de teléfonos de forma simultánea con el fin de realizar llamadas en la red; también ofrece varias opciones que permiten realizar llamadas, estas pueden ser, de forma local y en todos los dispositivos conectados a una radio base (PSTN), por voz ip (VoIP) o una red digital de servicios integrados (RDSI) (quarea, 2015). La función utilizada en este proyecto es VoIP.

Voz IP(VoIP): Es un conjunto de dispositivos y software que permiten que la señal de voz pueda usar el protocolo ip, el cual es el protocolo de internet que permite realizar llamadas de forma fluida (Daniel, 2016).

4.1.4. Ubuntu

Es un sistema operativo de software libre, que además es una distribución de Linux basado en Debian, que en la actualidad funciona tanto en ordenadores de escritorio y en servidores. Este sistema operativo se orienta a la facilidad de uso, debido a esto se ha popularizado de forma general llegando a acaparar el 52 % de ordenadores con distribuciones de Linux. El hecho de ser un software libre le brinda muchas ventajas, ya que la empresa que patrocina este sistema operativo, la que es canonical, se beneficia de los desolladores de su comunidad y ofrece servicios de soporte y parches de seguridad. En el desarrollo de este proyecto utilizamos la versión 12.04, ya que es la recomendada por el fabricante del hardware utilizado, y es la que ofrece mejor compatibilidad (ubuntu-co, 2018). se puede realizar la instalación de este sistema operativo en un dispositivo se requieren unos requisitos mínimos que permitan su funcionamiento sea óptimo, los cuales son:

- Procesador x86 a 1 GHz.

- Memoria RAM de 512 MiB.
- Disco Duro de 15 GB (swap incluida).
- Tarjeta gráfica y monitor capaz de soportar una resolución de 800×600.
- Lector de CD-ROM, puerto USB o tarjeta de red. (Carlos, 2014)

Estos requisitos los podemos encontrar en la gran mayoría de dispositivos del mercado, lo que lo hace factible de utilizar y ser económico.

4.1.5. Radio Definido por Software (SDR)

SDR o radio definido por software, se les llama a los sistemas capaces de realizar las funciones inalámbricas o de radar; estos pueden ser configurados a través de software, de tal forma que pueda trabajar a distintos rangos de frecuencia y se puedan adaptar en distintos protocolos lo que implica una gran ventaja a la hora de cambiar de protocolo; esto ayuda a minimizar costes, permite diseñar y probar modulaciones, lo que lo vuelve una herramienta extremadamente útil. Uno de los dispositivos más comunes que realizan SDR es el usrp, el que fue utilizado en la realización de este proyecto (eqeng, 2010).

4.1.6. USRP (Universal Software Radio Peripheral)

Es un dispositivo de radio definido por software, que permite realizar diseños de radios, crear prototipos de sistemas de telecomunicaciones, utilizar diferentes estándares de telecomunicaciones conocidos y propios mientras esté dentro de las capacidades del dispositivo. Esta herramienta es configurable a través de software y sus procesadores se encuentran basado en host de una FPGA y terminales frontales de RF. Estos dispositivos tienen una gran variedad de gamas dependiendo la necesidad (National Instruments, 2019). En este proyecto se usó el USRP N210, ya que se adapta a los requerimientos de este trabajo

y se contaba con acceso a este equipo por parte del laboratorio de telecomunicaciones de la Universidad de la Costa.

USRP N210: Este dispositivo tiene una capacidad de canal máxima de procesamiento hasta 100 Mb/s en ambas direcciones de la antena, a su vez ofrece buen procesamiento eficiente con respecto a la latencia. En la página oficial que ofrece el producto podemos encontrar sus características, las que son (National Instruments, 2019):

- El USRP N210 proporciona capacidad de procesamiento de alto ancho de banda y alto rango dinámico.
- El USRP N210 está diseñado para aplicaciones de comunicaciones exigentes que requieren este tipo de desarrollo rápido.
- La arquitectura del producto incluye un Xilinx® Spartan® 3A-DSP 3400 FPGA, 100 MS / s dual ADC, 400 MS / s dual DAC y conectividad Gigabit Ethernet con el fin de transmitir datos hacia y desde procesadores host.
- Un diseño modular permite que el USRP N210 funcione de CC a 6 GHz, mientras que un puerto de expansión permite que múltiples dispositivos de la serie USRP N210 se sincronicen y usen en una configuración MIMO.
- También se puede usar un módulo GPSDO opcional que permita disciplinar el reloj de referencia USRP N210 dentro de 0.01 ppm del estándar mundial de GPS.
- El USRP N210 puede transmitir hasta 50 MS / s desde y hacia aplicaciones host.
- Los usuarios pueden implementar funciones personalizadas en el tejido FPGA, o en el softcore RISC de 32 bits incorporado.

- El USRP N210 proporciona un FPGA más grande que el USRP N200 que funcione en aplicaciones que requieren recursos adicionales de lógica, memoria y DSP.
- El FPGA también ofrece el potencial de procesar hasta 100 MS / s en las direcciones de transmisión y recepción.
- El firmware FPGA se puede recargar a través de la interfaz Gigabit

4.2. Revisión del estado actual de la literatura

En referente a este trabajo se ha realizado una investigación donde se identificaron tres ejes temáticos que enmarcan los referentes teóricos y de conocimiento actual, que son los siguientes: Radio definido por software, Sistemas basados en OpenBTS y emulación de sistemas de telefonía celular.

Con el fin de realizar una búsqueda exhaustiva y que dé a conocer de manera clara los distintos trabajos realizados en los ejes temáticos identificados, se utilizó la herramienta SCOPUS para realizar una búsqueda sistemática de información relevante para esta investigación. Además de los trabajos relevantes encontrados, en la búsqueda de información con la herramienta se analizaron parámetros como la cantidad de artículos relacionados al eje temático por año, los países con más publicaciones del eje temático investigado y los autores con más contribuciones en la temática revisada; se tomaron datos desde el año 2000 hasta la actualidad 2020.

4.2.1 Tendencias en Radio Definido por Software

Con el fin de realizar el análisis de información en este eje temático se utilizó la cadena de búsqueda "software defined radio", en la herramienta scopus. Producto de esa búsqueda se tiene en la figura 1 los "autores más relevantes en la temática", en la figura 2 "los países donde más trabajos se han realizado de esta temática" y en la figura 3 trabajos relevantes

desde el año 2000 hasta enero del 2019

Documents by author

Compare the document counts for up to 15 authors.

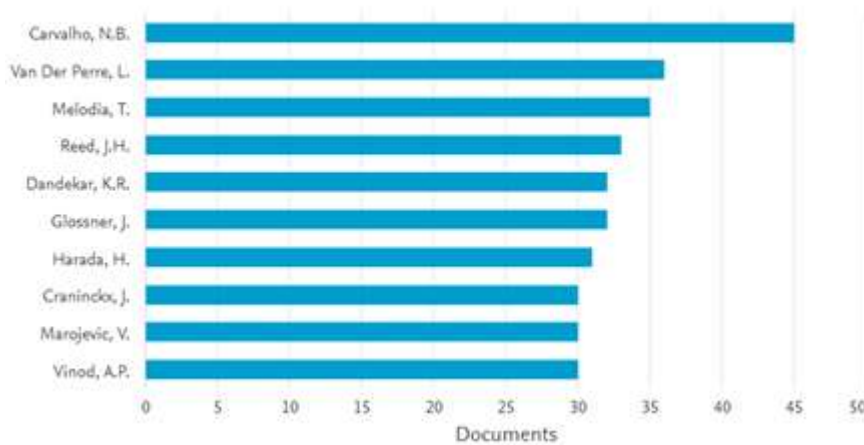


Figura 1 Autores más relevantes en la temática de Radio definido por software (fuente propia).

En la figura 1, vemos la cantidad de artículos publicados por diversos autores acerca del tema, tales son: Carvalho, N.B, Van Der Perre, L, Melodia, T. En los trabajos: “A fully reconfigurable software-defined radio transceiver in 0.13 μ m CMOS”, “Software-defined underwater acoustic networks: Toward a high-rate real-time reconfigurable modem”, “Designing and testing software-defined radios”.

Documents by country or territory

Compare the document counts for up to 15 countries/territories.

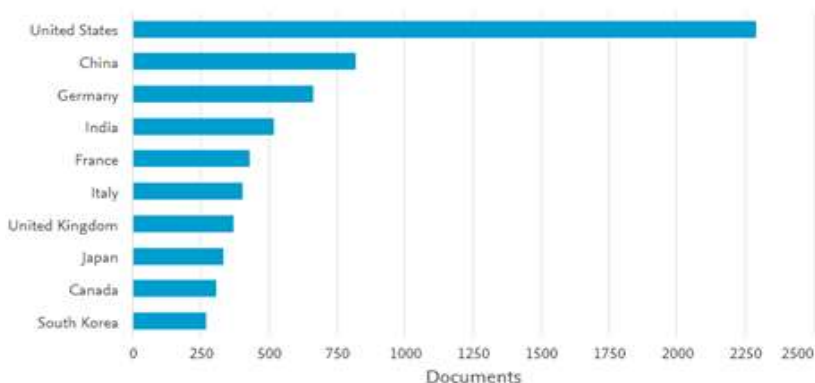


Figura 2 Cantidad de publicaciones por países en la temática de Radio definido por software (fuente propia).

En la figura 2, se observan los países con más publicaciones, encabezando el ranking se encuentra Estados Unidos, el cual es el país pionero en la investigación de este tema.

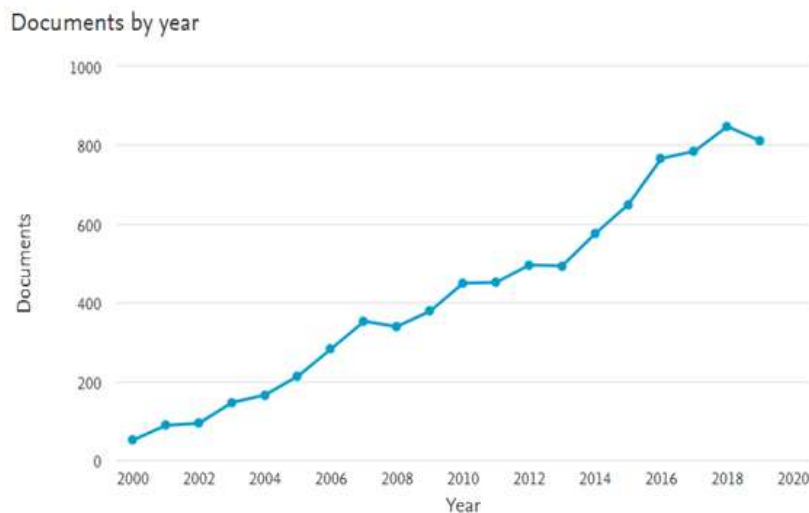


Figura 3 Documentos publicados por año en la temática de Radio definido por software (fuente propia).

En la figura 3, se observan la cantidad de publicaciones que se han realizado a través de los años en referente al tema, podemos concluir de la gráfica que hay un creciente interés en la temática.

Conforme a los trabajos revisados en esta temática es de destacar el titulado “**A Survey of 5G Network: Architecture and Emerging Technologies**” en donde se realiza una encuesta que recopila e informa los avances modernos en cuanto a la tecnología de comunicaciones 5G, hace un repaso histórico sobre los estándares usados en el pasado, la evolución de cada uno de estos y cómo se da evolución de la tecnología 5G; dentro del trabajo se realizan pruebas de los estándares, donde gran parte de estas estándares se realizaron usando radio definido por software. Debido a esto es uno de los artículos más citados en referente a esta temática, y muestra hasta qué punto pueden evolucionar los trabajos basados en 2G (Gupta, A., & Jha, R. K. 2015).

Otro documento que se consideró relevante es el titulado “**An Experimental Channel Capacity Analysis of Cooperative Networks Using Universal Software Radio Peripheral (USRP)**” el que analiza la comunicación cooperativa (CC) sobre comunicación directa (DC); a diferencia de la mayoría de trabajos este realiza un análisis basado en hardware en un

entorno de laboratorio; utilizando dos USRP se envían mensajes de texto a través de vía directa y modulada por M-PSK, con respecto a esto se determinó que: la ganancia de capacidad de canal promedio de CC sobre CD es 14.44% en BPSK, 14.59% en QPSK y 14.75% en 8PSK, respectivamente (Shujaat Ali Khan Tanoli, 2018).

En el trabajo titulado “**Low Altitude UAV Propagation Channel Modelling**” se da a conocer la eficiencia de la propagación de un canal que se comunica con un UAV (Vehículo Aéreo no Tripulado); este canal trabaja a 5.76 GHz y 1.817 GHz, bajo el estándar 802.11a. Con el fin de realizar las mediciones a distintas alturas y determinar el desvanecimiento de la señal; se usó un USRP en la comunicación, un hexacoptero con UAV, y el software Matlab que permite el análisis; resultando que bajo los 9 metros tenía un desvanecimiento de 1,23 dB y 2,15 dB respectivamente, esto debido a la zona de Fresnel (Xuesong Cai, 2017).

El trabajo titulado “**Análisis de desempeño del estándar IEEE 802.11AF en TVWS mediante la implementación en una plataforma de radio definido por software**” describe el diseño e implementación de un sistema WI-FI TVWS (TV White space); este trabajo tiene el objetivo de utilizar el estándar IEEE 802.11AF sea capaz crear una red inalámbrica capaz de transmitir señal de televisión en el rango de espectro de 54 MHz hasta 790 MHz; se realizar este trabajo se utilizaron 2 USRP-2920 y herramientas de software como Matlab y GND radio; se determinó que la distancia apropiada de comunicación son 12 m con el equipo utilizado, y aconsejan utilizar un equipo más potente, un ejemplo puede ser el USRP 2945 o 2955. (Alcocer Erazo, Jonathan David. 2019).

Finalmente, en el artículo “**Red de comunicaciones móviles mediante radio definida por software y USRP**” se explica el despliegue de un sistema GSM de radio definido por softwares usando un USRP, OpenBTS V5.0 que ofrecer servicios de voz y mensajes; esto con el fin de brindar servicios de telefonía móvil en caso de desastres naturales y realizar rescate

de víctimas de forma más fácil. Con el fin de realizar las pruebas se usaron los siguientes equipos: SPECTRAN HF-60105, una antena logarítmica de 45 dBi, un dispositivo GPS de marca GARMIN GPSMAP 62s, un USRP con antenas de 3 dBi omnidireccionales y una potencia de 10 dBm. En la realización de las pruebas se colocó el equipo USRP a una altura de 11 metros en un espacio abierto, y se tomaron varias muestras a diferentes distancias y obteniéndose una cobertura máxima de 220 m (Muñoz, Orozco, & Ramos, 2018).

4.2.2 Tendencias en Sistemas basados en OpenBTS

El análisis de este eje temático utilizó la cadena de búsqueda (OpenBTS AND systems), con la herramienta scopus; conforme a esta búsqueda se tiene que en la figura 4 se ven los “autores más importantes”, en la figura 5, se aprecian los países donde más trabajos se ha investigado esta temática y en la figura 6. se identifican los trabajos arrojados por la búsqueda desde el año 2010 hasta enero del 2019.

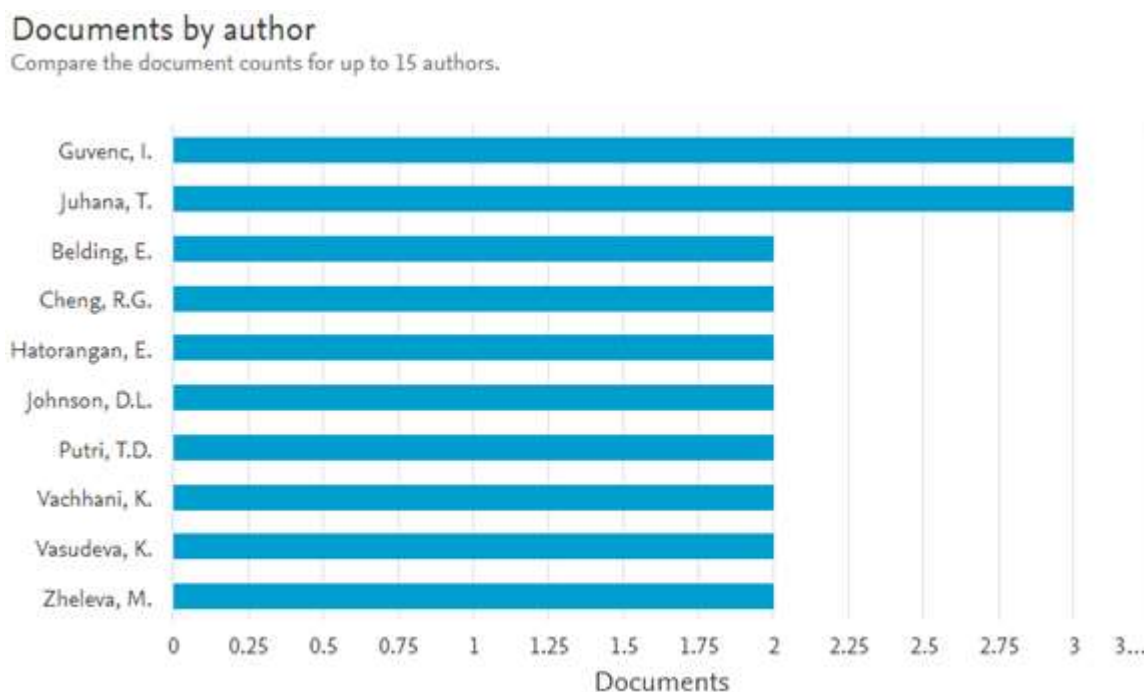


Figura 4 Autores más importantes en la temática de OpenBTS (fuente propia).

En la figura 4, vemos la cantidad de artículos publicados por diversos autores acerca del tema, tales son: Guvenc, I, Juhana,T,Belding E, en los trabajos : Kwiizya: Local cellular network services in remote áreas, UAV-based GSM network for public safety communications, Mobile phone auto registration to OpenBTS-based cellular network in disaster situation.

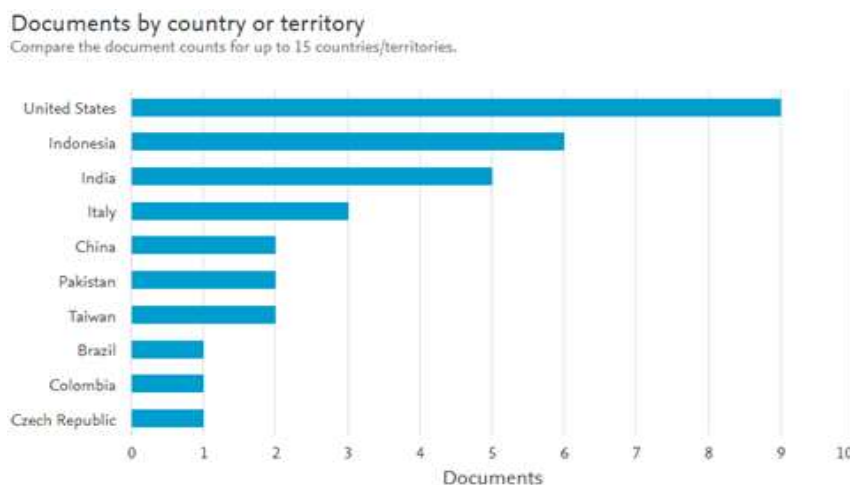


Figura 5 Países con más publicaciones en la temática de OpenBTS (fuente propia).

De la figura 5, se determina que el país con más publicaciones, encabezando este ranking, es los Estados Unidos.

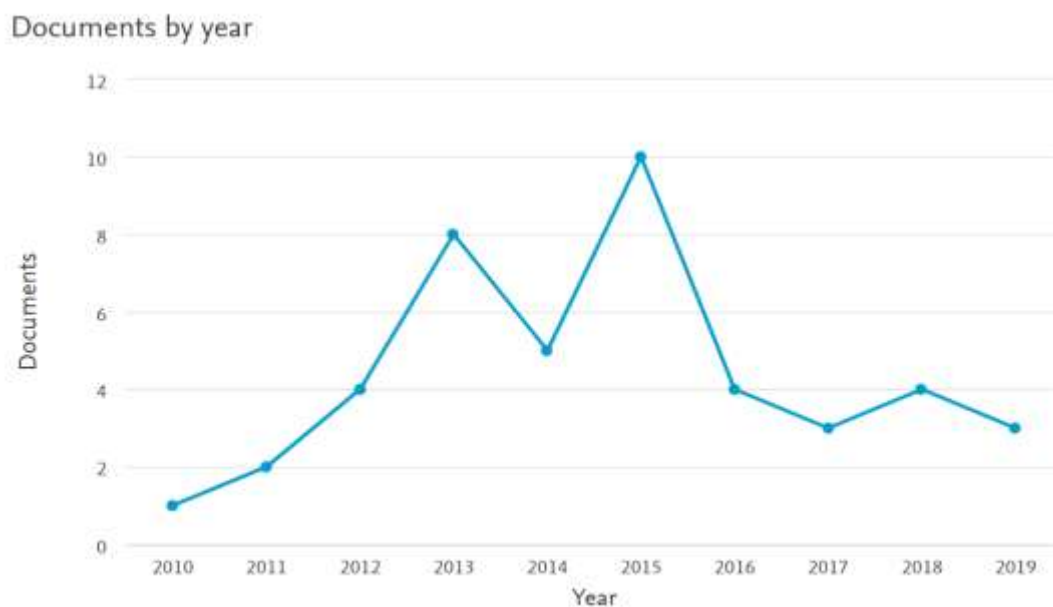


Figura 6 Documentos por años en la temática de OpenBTS (fuente propia).

En la figura 6, se ve la cantidad de publicaciones que se han realizado a través de los años en referente al tema, concluyendo de la gráfica que el tema sigue vigente en la actualidad.

Conforme a la revisión bibliográfica dentro de los artículos más relevantes se encuentra el trabajo **“Reconfigurable Cellular GSM Network using USRP B200 and OpenBTS for Disaster-Hit Regions”** el cual da a conocer una solución funcional de conectividad, a una red GSM, en caso de cataclismos. Se configura un radio USRP de bajo costo, que trabaja bajo el estándar GSM usando OpenBTS; se maneja el tráfico de mensajes y voz a través Asterisk, usando una frecuencia de 10 MHz; en la realización de las pruebas se configuraron los números telefónicos de cada celular dentro de la red y se comprobó que efectivamente enviaran mensajes de manera correcta; en pruebas de llamadas, estas se realizaron tipo “echo” con el fin de comprobar el estado de la conexión, lo que dio una pérdida de -50dB en los móviles. En este artículo encontramos una gran similitud con nuestro proyecto, lo que demuestra el gran interés que hay en referente al tema. (Kinjal Aggrawal, 2017)

Otro trabajo relevante titulado **“OpenBTS Based Microtelecom Model:A SocioEconomic Boon To Rural Communities”** plantea un modelo de negocio, el cual cubre una necesidad innata en el ser humano, la comunicación. Da a conocer cómo implementar un sistema de comunicación de bajo costo en zonas rurales, sin acceso a redes móviles, usando un software llamado OpenBTS y Asterisk; en la radio base se usó un radio USRP de baja potencia con el estándar GSM. Se realizó la instalación de este sistema en la comunidad CEN dando excelentes resultados (Prasannan, y otros, 2013).

El trabajo titulado **“Implementation of USRP (Universal Software Peripheral Radio) as OpenBTS for Quadruple Play Services”** expone la capacidad y el alcance que pueden llegar a tener los sistemas basados en OpenBTS, con la llegada de nuevas tecnologías que

permiten aumentar el alcance de las telecomunicaciones a sitios remotos a costos moderados, plantea el uso de herramientas tales : OpenBTS, USRP, Asterisk PBX, GNU Radio; que permitan implementar sistemas de comunicación con el estándar GSM y voz ip; logrando llevar tecnologías a sitios incomunicados y poco atractivos a las compañías de telefonía celular (Hafidudin, 2019).

En el documento titulado “**Towards Designing a Microcell Base Station using a Software-Defined Radio Platform**” se estudia el rendimiento de una antena en tecnología microstrip diseñado especialmente para aplicaciones de micro celulares en un contexto real de la telefonía móvil, en la realización de las pruebas de dicha antena se utilizó el protocolo GSM, usando la herramienta OpenBTS y el USRP N210; esto se trabajó en la banda de los 6 GHz con el fin medir la potencia de la señal, teniendo como métrica el indicador RSSI (Received Strength Signal Intensity); los móviles en los que se realizó un modelamiento en 3D de la propagación a diferentes frecuencias, con el fin de determinar a cuáles frecuencias se obtuvo mejor recepción (Zakaria Er-reguig, 2019).

Finalmente, el artículo “**SDR Based Indoor Localization Using Ambient WiFi and GSM Signals**” se plantea un método que determine la posición de un dispositivo móvil en el interior de un recinto, utilizando una red GSM y WIFI; con el fin de desarrollar este método se utilizaron diversas herramientas de software como: GNU Radio, OpenBTS y Wireshark, sobre un equipo USRP. La detección se realiza con la técnica del mínimo cuadrado obteniendo señales de GSM y de WIFI, con resultados satisfactoriamente precisos en la ubicación de los dispositivos móviles (Varun Nambiar, 2016).

4.2.3 Tendencias en herramientas para modelamiento e implementación de un sistema de telefonía celular móvil.

En el análisis de información en este eje temático se usó la cadena de búsqueda (tools AND phone AND (modeling OR implement)); similarmente en la figura 7 se ven “autores más importantes”, la figura 8 ven los países donde más trabajos se han realizado de esta temática y la figura 9. ven los trabajos más relevantes desde el año 2000 hasta enero del 2019.

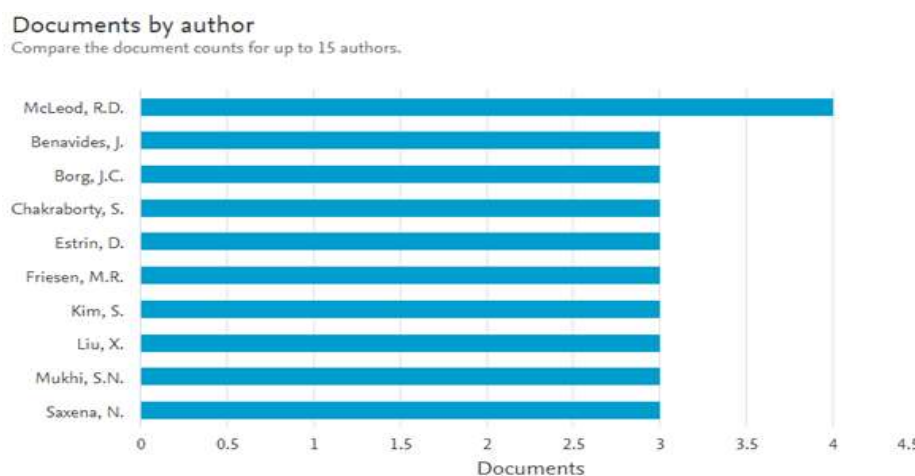


Figura 7 Autores más relevantes en la temática de modelamiento y/o implementación de sistemas de telefonía celular móvil (fuente propia).

En la figura 7, vemos la cantidad de artículos publicados por diversos autores acerca del tema, tales son: McLeod, R.D, Benavides, J, Borg.J.C. En los trabajos: Smartphone and vehicular trajectories as data sources for agent-based infection spread modelling, Smartphone technologies for social network data generation and infectious disease modeling, Cameraphones as new design tools.

Documents by country or territory
Compare the document counts for up to 15 countries/territories.

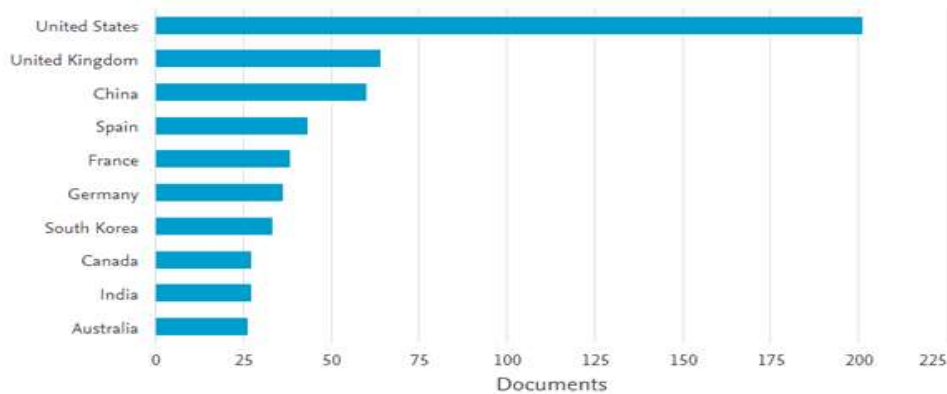


Figura 8 Publicaciones por países en la temática de modelamiento y/o implementación de sistemas de telefonía celular móvil (fuente propia).

En la figura 8, se tienen los países con más publicaciones, encabezando el ranking se encuentra Estados Unidos, este país es el que muestra mayor interés en el desarrollo del tema.

Documents by year

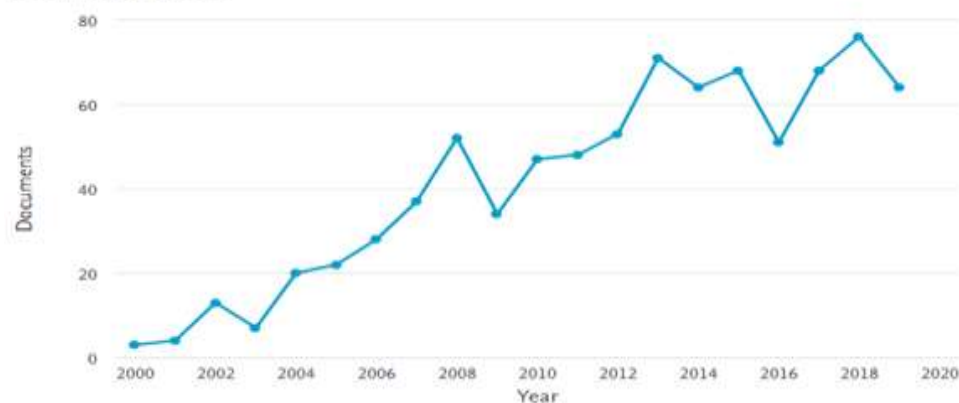


Figura 9 Publicaciones por año en la temática de modelamiento y/o implementación de sistemas de telefonía celular móvil (fuente propia).

La figura 9, muestra la cantidad de publicaciones que se han realizado a través de los años en referente al tema, la que demuestra un crecimiento.

Una vez realizada esta revisión, dentro de los documentos más importantes se encuentra **“GSM system implementation for academic purposes”** el cual expone el desarrollo de un proyecto en el que se utiliza una USRP con el fin de establecer comunicación en una zona de laboratorios, esto en la universidad pontificia bolivariana de Medellín; se utilizó el sistema

operativo basado en Linux GNU para realizar el desarrollo del proyecto, un radio USRP, dos teléfonos móviles, el software OpenBTS y Asterisk para configurar y realizar el tráfico de voz y mensajes. Este proyecto dio como resultado una satisfactoria comunicación en los distintos laboratorios (Jorge E. Pulgarín, 2013).

Otro trabajo que se destaca es el titulado **“Implementación de un receptor ONE SEG utilizando la plataforma de radio definido por software USRP para el estándar ISDB-Tb”** realizado por la Universidad de las Fuerzas Armada, donde se propuso implementar un receptor de televisión digital terrestre bajo el estándar ISDB-TD, con el fin que funcione con un USRP, se implementó en el dispositivo un algoritmo de recepción que permite recibir bandas estrechas. Se utilizaron distintas herramientas que permiten el análisis de la recepción como lo fueron el USRP y el software Matlab; este análisis se determinó que a valores superiores a -90 dB las constelaciones no varían en gran magnitud, dado así una buena recepción (Garzón Gaona, 2018).

En el trabajo titulado **“Modeling and Analysis of Two-tier MIMO Heterogeneous Cellular Network”** este trabajo proporciona un análisis más preciso del límite de rendimiento de las redes celulares de varios niveles; estos aportes se pueden resumir en el proceso de agrupación de Poisson para modelar la red celular MIMO de dos niveles, se buscó la transformada de Laplace de la interferencia, la probabilidad de cobertura y La eficiencia espectral promedio en la red celular MIMO de dos niveles. En el trabajo se utilizan herramientas matemáticas que argumentan las eficiencias del método planteado, este trabajo da pie a realizar una implementación de un sistema MIMO optimizado (Jianchao Zhenga, 2017).

Finalmente, el documento **“Development and pilot testing of a text message vaccine reminder system for use during an influenza pandemic”** plantea el modelamiento de un

sistema de telefonía celular que brinda una solución ante una pandemia de gripa; en este caso puntual se necesita inyectar dos dosis que logren la inmunidad al virus; en esto se utilizó un software creado por la compañía CDC, el que consiste en una serie de alertas y recordatorios a las personas a través de mensajes de texto; esto se sometió a prueba la efectividad del sistema y se obtuvo que: el 92% consideró el sistema sería útil durante una pandemia; el 68% de los encuestados consideró la información contenida en los mensajes era informativo (Jonathan D. Lehnert, 2018).

5. Metodología

5.1. Tipo de Estudio

Se reproduce una estación base de telefonía celular móvil a través de herramientas libres, consistentes en una plataforma de radio definido por software (OpenBTS) y un hardware de radiofrecuencia (USRP). En el aprendizaje e investigación se contempla el uso de dos tipos de estudio: Exploratorio y descriptivo.

Los estudios exploratorios se evidencian en la obtención de información, la formulación del problema y la revisión documental sobre herramientas de radio definido por software que permita aprendizaje de telefonía celular móvil. El trabajo se llevará a cabo en 4 Fases descritas a continuación:

Tabla 1 *Descripción del desarrollo de las fases del proyecto*

Fases	Actividad	Indicador	Resultado
--------------	------------------	------------------	------------------

Identificación de las características del prototipo de OpenBTS a desarrollar.	Realizar una revisión del estado del arte de los sistemas de telefonía celular móvil GSM.	Lista de componentes de un sistema de telefonía móvil celular para el escenario de aplicación del proyecto.	Documento de revisión de la literatura con relación a un sistema de telefonía celular
	Identificar los principales componentes de un sistema de telefonía celular móvil.		
Caracterización del sistema a desarrollar	Realizar una revisión de las herramientas de software para el aprendizaje e investigación de redes de telefonía celular móvil GSM.	18 artículos revisados	Documento de caracterización con diagrama de bloques del sistema OpenBTS
	Realizar una revisión de casos de estudio similares relacionados a diseño de herramientas de radio definido por software para simular		

una red de telefonía
celular móvil.

Identificar los
bloques funcionales
de la herramienta de
radio definido por
software que permita
simular un sistema de
telefonía móvil celular
GSM.

Número de
elementos
seleccionados en la
implementación del
OpenBTS.

Caracterizar el
hardware y software a
utilizar en cada uno de
los bloques
funcionales de la
herramienta de radio
definido por software
que permita simular
un sistema de
telefonía móvil celular
GSM.

Implementación
del prototipo de

Analizar la
compatibilidad en
cuanto a software y
hardware en los

PC con Linux
instalado

Documento de guía
metodológico de

OpenBTS a desarrollar.	equipos de cómputo a utilizar.	instalación y uso de una OpenBTS
<hr/>		
Instalar un sistema operativo basado en Linux que permita la implementación de las herramientas de software a utilizar.		
<hr/>		
Descargar e instalar las librerías básicas necesarias para la herramienta de radio definido por software que permita simular un sistema de telefonía móvil celular GSM.		OpenBTS implementado
<hr/>		
Instalar los drivers UHD necesarios que permitan el funcionamiento de la herramienta de radio definido por software que permita simular un sistema de		

telefonía móvil celular
GSM.

Realizar la
configuración de radio
con el paquete
OPENBTS.

Realizar la
instalación final de
OPENBTS con el
radio conectado al
computador.

Ejecutar
OPENBTS y cada
componente del
paquete con el radio
conectado.

Instalar y
configurar un sistema
de conmutación
basado en
ASTERISK.

Herramienta
Asterik instalada

Verificación del
prototipo de
OpenBTS a
desarrollar.

Realizar pruebas
de envío de mensajes
SMS entre usuarios
distintos.

Pruebas de SMS
realizadas

Documento de
análisis de pruebas de
funcionamiento de

Realizar pruebas de envío de mensajes SMS entre la estación base y los usuarios.		prototipo de sistema OpenBTS
<hr/>		
Realizar pruebas de llamadas en los usuarios conectados a la herramienta de radio definido por software que permita simular un sistema de telefonía móvil celular GSM.	Pruebas de llamadas realizadas.	
Analizar los resultados obtenidos en las pruebas realizadas con la herramienta de radio definido por software que permita simular un sistema de telefonía móvil celular GSM.		

Realizar la
documentación del
proyecto y un manual
de operación de la
herramienta de radio
definido por software
que permita simular
un sistema de
telefonía móvil celular
GSM.

Nota. Descripción de las actividades, indicadores y los resultados en el desarrollo de cuatro fases del proyecto (fuente propia).

5.2. Identificación y caracterización de la herramienta OpenBTS a desarrollar.

La caracterización de este proyecto está asociada a implementar una herramienta dirigida al programa de Ingeniería Electrónica de la Universidad de la Costa que permita profundizar el entendimiento de los sistemas de telefonía móvil celular y además, brindar soluciones a futuro dentro de una arquitectura propia de transmisión de información. Se requiere una herramienta para ser usada y operada en los laboratorios del área de Sistemas de Telecomunicaciones. Con el fin de llevar a cabo la OpenBTS se realizó una caracterización inicial del sistema de acuerdo con la revisión bibliográfica, comparándolo con un sistema de telefonía celular móvil GSM tradicional, la Figura 10 muestra esta comparación.

La figura 10 identifica los elementos funcionales de una OpenBTS que emula el comportamiento de un sistema GSM tradicional, los cuales fueron definidos en el capítulo 2. La figura 10 muestra, además, las similitudes de la OpenBTS y un sistema GSM tradicional; en primera instancia está que los dos sistemas se conectan a un SM (estación móvil) que

prestan los servicios de red. Además, el OpenBTS maneja las mismas frecuencias que el sistema GSM tradicional.

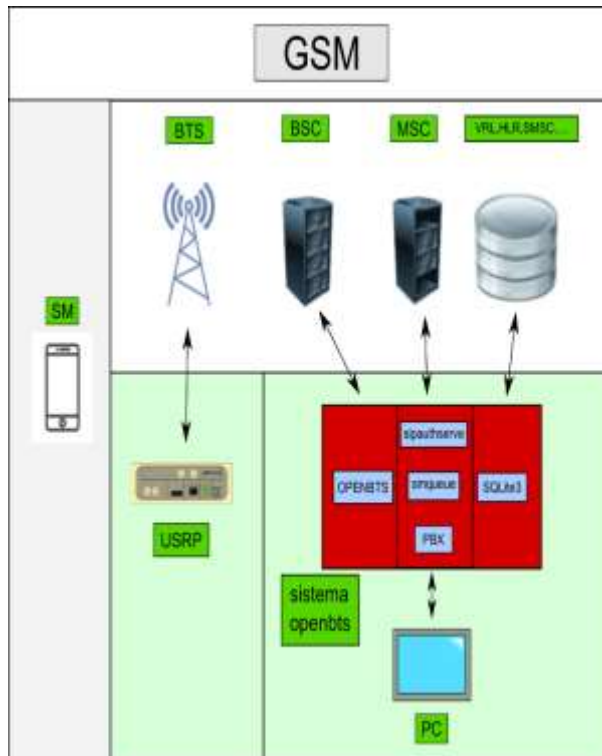


Figura 10 Comparación del sistema OpenBTS con un sistema GSM tradicional (Manuel Quiñones, 2015).

El sistema OpenBTS comienza a diferenciarse de un sistema GSM desde la BTS ya que esta es más simplificada; en un sistema GSM tradicional la BTS posee los radios transceptores y sistema radiante (antenas y cableado RF) que se encargan de transmitir la señal GSM; mientras una BSC se encarga de Administrar la red; la MSC se encarga del encaminamiento y conmutación correspondiente de los servicios móviles (mensajes, llamada y registros) tomando decisiones por medio de Bases de datos de control o auxiliares. Estas son reemplazadas en el sistema OpenBTS por un radio USRP que se encarga transmitir la señal GSM; el software OpenBTS incorporado en una PC que funciona a modo de servidor, es un sistema que posee varias herramientas que se complementan y que funcionan como la BSC, MSC y Base de datos de un sistema GSM tradicional (Manuel Quiñones, 2015).

El funcionamiento interno del sistema OpenBTS se ve la figura 11, la que detalla esta información.

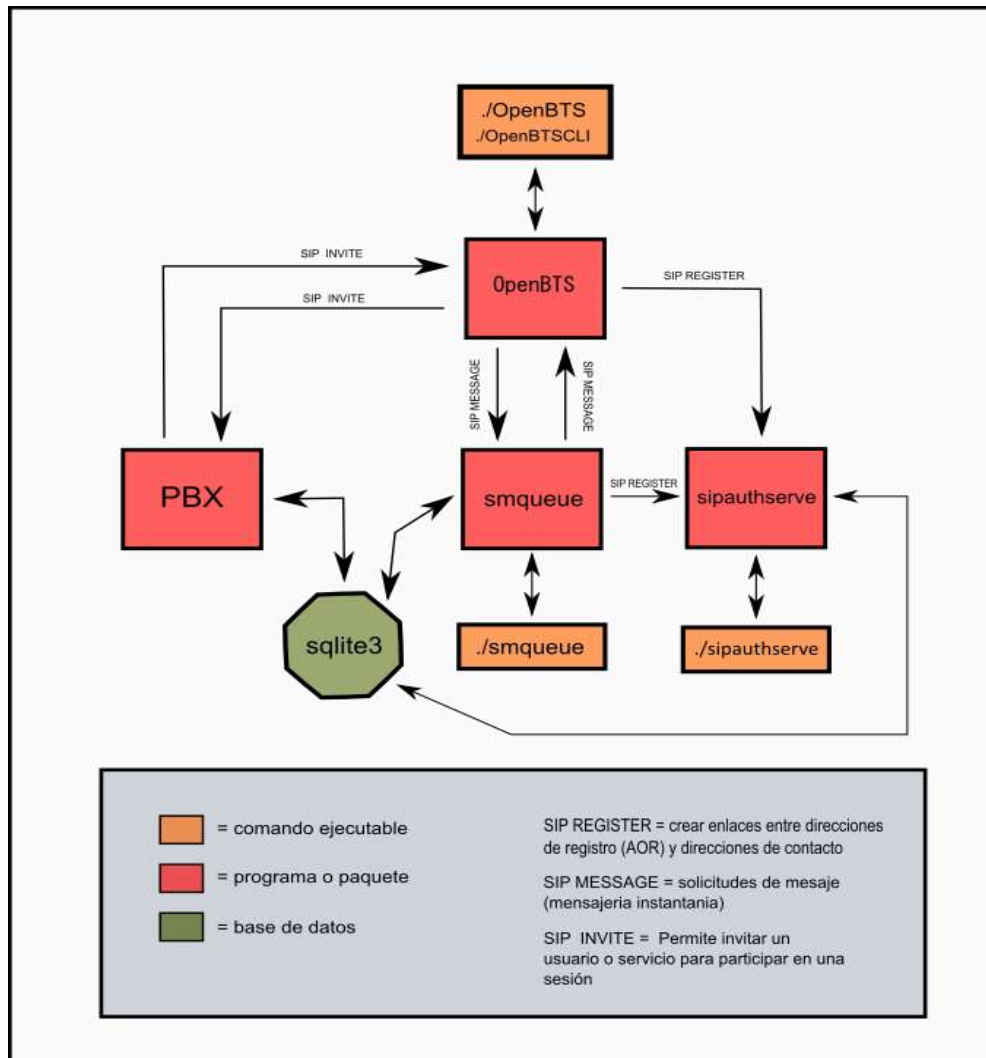


Figura 11 Esquema de operación del sistema OpenBTS (Manuel Quiñones, 2015).

La figura 11 expone el funcionamiento a través de bloques del sistema OpenBTS al momento de simular la red GSM; estos bloques se describen a continuación:

Bloque OpenBTS, se encarga de la configuración general de la red GSM, teniendo como principales funciones las siguientes:

- Programación del radio a utilizar (establecer ganancia de transmisión, potencia, frecuencia de operación, y otros).

- Establecer la comunicación entre los equipos móviles que harán parte de la red,
- Configuración de la red móvil celular.
- Permitir el enlace con los otros bloques con el fin de que puedan ofrecer sus servicios a la red.

Bloque de servicio de autenticación y registro (sipauthserve), suministra al sistema OpenBTS el servicio de autenticación y registro de usuarios móviles. En su operación, recibe todas las solicitudes de autenticación de cada equipo móvil conectado, verifica su información y posteriormente la guarda en la base de datos.

Bloque de servicio de mensajería instantánea (smqueue), brinda el servicio de mensajería instantánea a los dispositivos móviles. A través de éste los equipos conectados a la red pueden enviar mensajes de texto a todos los usuarios. Para su operación, consulta la base de datos interna, que permite identificar la procedencia de los mensajes.

Bloque de conmutación (PBX), se encarga del servicio interconexión de llamadas; permite la transmisión y recepción de voz entre equipos dentro de la red por medio de la base de datos.

Bloque de base datos (sqlite3), almacena y asegura la información de los equipos registrados dentro de la red; también permite que los otros bloques accedan a ella con el fin poder realizar de manera correcta la comunicación y respuesta a los móviles conectados a la red.

Con el objetivo de que el diseño realizado se pueda llevar a cabo, la OpenBTS desarrollada cuenta con las siguientes herramientas de hardware y software, que son asociadas a los bloques funcionales descritos anteriormente en las figuras 10 y 11.

En cuanto al componente de hardware de la herramienta, se utilizó el radio USRP N210, de la familia Ettus Research USRP TM que es un equipo versátil capaz de transmitir y recibir diferentes frecuencias configuradas por el usuario a través de una plataforma de hardware común; el USRP es flexible y reutilizable en las implementaciones de soluciones, investigación de señales y exploración de señales inalámbricas (Ettus, 2020).

El USRP tiene varias categorías, que definen la forma y configuración para operar, que son:

Serie Bus (B): conectado a una computadora host a través de una conexión USB

Serie Network (N): conectada a una computadora host a través de una conexión Ethernet

Serie de alto rendimiento (X): la conexión puede ser Ethernet o una conexión PCI-Express x4

Serie incrustada (E): diseñada con el fin de ejecutarse de manera independiente (sin una computadora host).

De acuerdo con las categorías dadas se escoge la serie Network (N) por la conectividad Gigabit Ethernet que permite transmitir datos hacia y desde los procesadores host de forma más directa, además porque este tipo de conexiones cubre más distancia sin que se degrade la calidad de la señal. De la serie N se elige la N210 por su compatibilidad más estable al software y sistema operativo a utilizar. A continuación, se ven sus especificaciones en la figura 12:

Spec	Typ.	Unit	Spec	Typ.	Unit
ENERGIA			RF RENDIMIENTO (W/ WBX)		
DC entrada	6	V	SSB/LO supresión	35/50	dBc
Consumo de corriente	1.3	A	Ruido de fase (1.8 Ghz)		
w/ WBX tarjeta hija.	2.3	A	10 kHz	-80	dBc/Hz
RENDIMIENTO DE CONVERSIÓN Y RELOJES			100 kHz	-100	dBc/Hz
ADC frecuencia de muestreo	100	MS/s	1 MHz	-137	dBc/Hz
ADC resolución	14	bits	Potencia de salida	15	dBm
ADC banda ancha SFDR	88	dBc	IIP3	0	dBm
DAC frecuencia de muestreo	400	MS/s	Figura de ruido del receptor	5	dB
DAC resolución	16	bits	FÍSICO		
DAC banda ancha SFDR	80	dBc	Temperatura de funcionamiento	0 a 55°	C
frecuencia de muestreo del host (8b/16b)	50/25	MS/s	Dimensiones (l x w x h)	22 x 16 x 5	cm
Precisión de frecuencia	2.5	ppm	Peso	1.2	kg
w/ GPSDO referencial	0.01	ppm			

Figura 12 Especificaciones de la USRP N210 (Ettus, 2020).

Unas de las cualidades del USRP N210 es su adaptabilidad con diversas tarjetas complementarias que lo hacen compatible a la hora de simular una red GSM. Una de esas tarjetas es la WBX 50-2200 MHz Rx/Tx (40 MHz). El WBX es un transceptor de ancho de banda amplio. Los LO (oscilador local) para las cadenas de recepción y transmisión operan independientemente, pero pueden sincronizarse con el fin de operar en modo MIMO (multiple inputs multiple outputs). La tarjeta WBX proporciona 40 MHz de capacidad de ancho de banda y es ideal para aplicaciones que requieren acceso a varias bandas diferentes dentro de su rango: 50 MHz a 2.2 GHz. Las áreas de aplicación de ejemplo incluyen comunicaciones móviles terrestres, radios de banda marítima y de aviación; estaciones base de telefonía celular, PCS y radios multibanda GSM; radares coherentes multi estáticos; sensor de redes inalámbricas; transceptor que cubre 6 bandas de aficionados; transmisión de TV; espacios en blanco; seguridad Pública; ISMO (National Instruments, 2020). Además, la tarjeta WBX 50-2200 MHz Rx/Tx (40 MHz) destaca por su amplitud de funciones y aplica también la actividad a realizar.

En la propagación de la señal, la USRP tiene conectores delanteros que permiten conectar antenas en las bandas de frecuencia de GSM900 y DCS1800; el sistema propuesto utiliza dos antenas Vert 900, Omnidireccionales con conectores SMA macho genéricas. La figura 13, la

figura 14 y la figura 15 a continuación se ve el radio USRP N210 utilizado, con la tarjeta WBX y las antenas.



Figura 13 Ettus N210 parte interna (fuente propia).

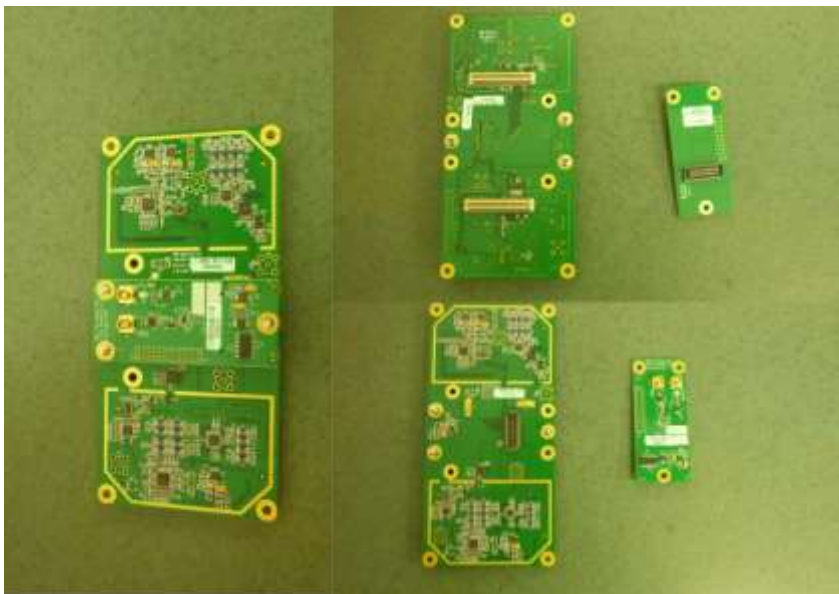


Figura 14 WBX 50-2200 MHz Rx/Tx (40 MHz) (fuente propia).



Figura 15 Ettus N210 armado y conexión (fuente propia).

Con relación a la USRP, esta cuenta con controladores que permiten la compatibilidad a los diversos tipos de sistemas, autodenominados UHD (controlador de hardware USRP), que es una biblioteca de espacio de usuario que se ejecuta en un procesador de propósito general (GPP); se comunica y controla toda la familia de dispositivos USRP de las series B, N y X. Estos envían y reciben muestras en una computadora dado que algunas USRP albergan un GPP interno y pueden ejecutar la codificación en el mismo radio, sin la necesidad de un proceso externo llevado a cabo en una computadora host (modo independiente). Los USRP son transceptores, lo que significa que pueden transmitir y recibir señales de RF. El UHD proporciona el control necesario utilizado para transportar muestras de forma de onda del usuario hacia y desde el hardware USRP, también controla varios parámetros (por ejemplo, frecuencia de muestreo, frecuencia central, ganancias, etc.) de la radio.

La mayoría de la base de código UHD es de código abierto, incluido el código que se ejecuta en el host, también el código dirigido al hardware USRP (FPGA y firmware de microcontrolador). El código de firmware y controlador UHD GPP está escrito en C / C ++ mientras que el código desarrollado en la FPGA (Field Programmable Gate Array) está escrito en Verilog. Existe una API C / C ++ que puede interactuar con otros marcos de

software, en el caso de GNU Radio, o un usuario puede simplemente crear aplicaciones de procesamiento de señales personalizadas directamente sobre la API UHD C / C ++ (Ettus, 2020).

En la figura 16 se observa la interfaz de hardware de un USRP y una computadora.

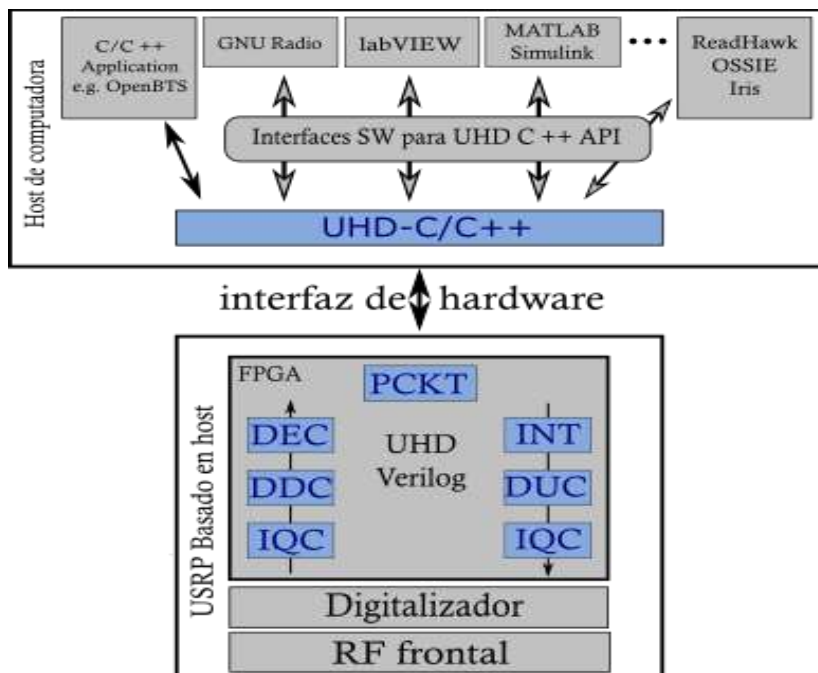


Figura 16 Diagrama de bloques para describir el funcionamiento del UHD de una USRP (Ettus, 2020)

En la realización de este proyecto se decidió utilizar el sistema operativo Ubuntu versión 12.04 lts de 64 bits, en lugar de otro sistema operativo ya que es el recomendado y ofrece un funcionamiento más estable según las pruebas realizadas por el desarrollador de la distribución de LINUX; además esta distribución es más compatible con el sistema OpenBT. Se instaló la versión de desarrollo v2.8 ya que es gratuita, a diferencia de otras versiones pagas, la que funciona de forma eficiente en Ubuntu versión 12.04 lts de 64 bits y no en versiones diferentes del sistema operativo por incompatibilidad de las librerías. En versiones superiores algunas librerías quedan obsoletas y en versiones más viejas no existen.

Una de las ventajas de este sistema operativo es que es híbrido, lo que lo hace compatible con otros sistemas y da la opción de usar un dispositivo personal (computador, raspberry pi, u

otro) con el fin de instalarlo con el OpenBTS, siempre y cuando cumpla las especificaciones técnicas mínimas del equipo o sea superior a ellas.

Basándose en las especificaciones técnicas que permiten tener Ubuntu 12.04 y teniendo en cuenta los requerimientos que pide OpenBTS las especificaciones serán las siguientes:

Tabla 2 *Especificaciones del equipo de cómputo que usa OpenBTS*

Hardware	Capacidad mínima
Memoria RAM	≥ 2 GB
Disco Duro	20 GB
Procesador	x64 >1 Ghz
Ethernet	1 gigabit
Tarjeta gráfica y monitor	800x600 resolución

Nota. Especificaciones del equipo de cómputo que usa OpenBTS que permiten tener Ubuntu 12.04 (fuente propia).

Si queremos que el OpenBTS pueda ejecutar llamadas se necesita una herramienta externa que complemente el sistema, con el fin de realizar la conmutación entre los usuarios. En este caso se usó Asterisk Real Time, la que es una distribución de Asterisk que es un software libre que proporciona las funcionalidades de una central telefónica. La arquitectura Asterisk Real Time se diferencia de otras por su flexibilidad y escalabilidad. Además, permite configurar aspectos del sistema en tiempo real y de forma dinámica a través del uso de una base de datos. Esto lo hace ser un complemento compatible con el sistema OpenBTS ya que se puede vincular a su base de datos (wikiasterisk, 2014). Debido al tema de compatibilidad y la versión del sistema operativo se usará la versión Asterisk 10.5.1

Con base a la caracterización de la herramienta realizada y a selección de los materiales anteriormente descritos, se procede en la siguiente subsección de la metodología a describir la implementación del Sistema OpenBTS.

5.3. Implementación de la OpenBTS para el Programa de Ingeniería Electrónica de la Universidad de la Costa.

Con el fin de realizar la implementación del sistema OpenBTS, una vez teniendo el equipo de cómputo configurado con el hardware necesario y con la versión de LINUX Ubuntu 12.04 lts, se siguieron los siguientes pasos.

5.3.1. Configuración de la herramienta de radio definido por software para el envío de mensajes SMS y llamadas de voz GSM.

En esta actividad se realizó la descarga e instalación de las dependencias básicas necesarias con el fin de que la herramienta de radio definido por software emule un sistema de telefonía móvil celular GSM. En esta parte se indica de qué manera se implementaron en la herramienta OpenBTS bloques como el de base datos, autenticación y registro y mensajería instantánea, los cuales se muestran en la figura 11 (repositorio.unicauca.edu.co, 2015).

En esta actividad se cuenta con la distribución Ubuntu 12.04 con todas las actualizaciones previamente instaladas; usando el modo de comandos, se ingresa a la terminal en modo “súper-usuario” y se instalan las dependencias que se encuentra en la tabla 3. Estas dependencias se deben instalar en orden y garantizar así, que la OpenBTS funcione correctamente; con el fin de obtener mayor información acerca del proceso de instalación de la OpenBTS, se puede acceder a la página oficial de OpenBTS, sin embargo, las instrucciones dadas son claras. En la tabla a continuación se detallan las dependencias instaladas en el orden, con su respectivo comando y la librería instalada.

Tabla 3 *Lista de dependencias para OpenBTS*

Comando	Librería, dependencia o programa.	Observación
sudo apt-get install	git	
sudo apt-get install	swig	
sudo apt-get install	cmake	
sudo apt-get install	doxygen	Al terminar se se debe ejecutar el comando <i>ldconfih</i>
sudo apt-get install	build-essential	
sudo apt-get install	libboost-all-dev	
sudo apt-get install	libtool	
sudo apt-get install	libusb-1.0-0	
sudo apt-get install	libusb-1.0-0-dev	
sudo apt-get install	libudev-dev	
sudo apt-get install	libncurses5-dev	
sudo apt-get install	libfftw3-bin	
sudo apt-get install	libfftw3-dev	
sudo apt-get install	libfftw3-doc	

sudo apt-get install	libc++unit-1.1	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	libc++unit-dev	
sudo apt-get install	libc++unit-doc	
sudo apt-get install	ncurses-bin	
sudo apt-get install	cpufrequtils	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	python-numpy	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	python-numpy-doc	
sudo apt-get install	python-numpy-dbg	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	python-scipy	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	python-docutils	
sudo apt-get install	qt4-bin-dbg	
sudo apt-get install	qtcreator	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	synaptic	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	libqt4-*	Al terminar se debe ejecutar el comando <i>ldconfig</i>
sudo apt-get install	qt4-doc	
sudo apt-get install	libqt4-dev	
sudo apt-get install	python-qt4	
“”	python-qt4-dbg	

“”	python-qt4-dev	
“”	python-qt4-doc	
“”	libfftw3-dev	
“”	libncurses5	
“”	libncurses5-dbgsym	
“”	libfontconfig1-dev	
“”	libxrender-dev	
“”	libpulse-dev	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	g++	
“”	automake	
“”	autoconf	
“”	python-dev	
“”	libusb-dev	
“”	fort77	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	libsdl1.2-dev	
“”	python-wxgtk2.8	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	git-core	
“”	ccache	
“”	python-opengl	Al terminar se debe ejecutar el comando <i>ldconfig</i>

“”	libgsl0-dev	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	python-cheetah	
“”	python-mako	
“”	python-lxml	
“”	qt4-dev-tools	
“”	libqwt5-qt4-dev	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	libqwtplot3d-qt4-dev	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	pyqt4-dev-tools	
“”	python-qwt5-qt4	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	wget	
“”	libxi-dev	
“”	gtk2-engines-pixbuf	
“”	r-base-dev	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	python-tk	Al terminar se debe ejecutar el comando <i>ldconfig</i>
“”	liborc-0.4-0	
“”	liborc-0.4-dev	
“”	libasound2-dev	
“”	python-gtk2	
“”	libzmq1	Al terminar se debe ejecutar el comando <i>ldconfig</i>

“”	libzmq-dev	
“”	python-requests	
“”	python-sphinx	
“”	libcomedi-dev	
“”	subversion	
“”	libgsm1-dev	
“”	libosip2-dev	
“”	libortp-dev	
“”	python-setuptools	
“”	python-dev	
“”	build-essential	
“”	curl	
	sqlite3	
	libsqlite3-dev	
	libreadline6-dev	
	libortp8	
	libortp-dev 3.2.1	<i>ldconfig</i>
	apache2	

subversion	
libapache2-svn	
git-svn	ldconfig

Nota. Lista de dependencias para OpenBTS, especificación de cada comando con su respectivo comando y la librería instalada a ejecutar (*fuentes propias*).

Una vez realizado este proceso, se continúa con la instalación de la herramienta de Asterisk. Posteriormente se instalan los drivers UHD necesarios en el funcionamiento de la herramienta de radio definido por software que permita simular un sistema de telefonía móvil celular GSM y permite ofrecer el soporte a los dispositivos UHD. En la instalación del UHD en la plataforma Linux (Ubuntu) se hace uso de la terminal de comandos en modo súper-usuario y primero, se direcciona a la carpeta home, luego se crea la carpeta “workarea-uhd” y se entra en ella, de la siguiente manera:

- *cd \$HOME*
- *mkdir workarea-uhd*
- *cd workarea-uhd*

Dentro de esta carpeta se usa el siguiente repositorio que permite instalar las librerías necesarias para los radios de la familia ETTUS; esto permitirá crear una carpeta llamada UHD; una vez creada, se accede a ella; lo anterior se logra usando las siguientes instrucciones:

- *git clone <https://github.com/EttusResearch/uhd>*

- *cd uhd*

Una vez en esta carpeta se introducen los siguientes comandos para la configuración e instalación de las dependencias necesarias:

- *git tag -l*

- *git checkout release_003_009_007*

- *cd host*

- *mkdir build*

- *cd build*

- *cmake ../*

- *make*

- *sudo make install*

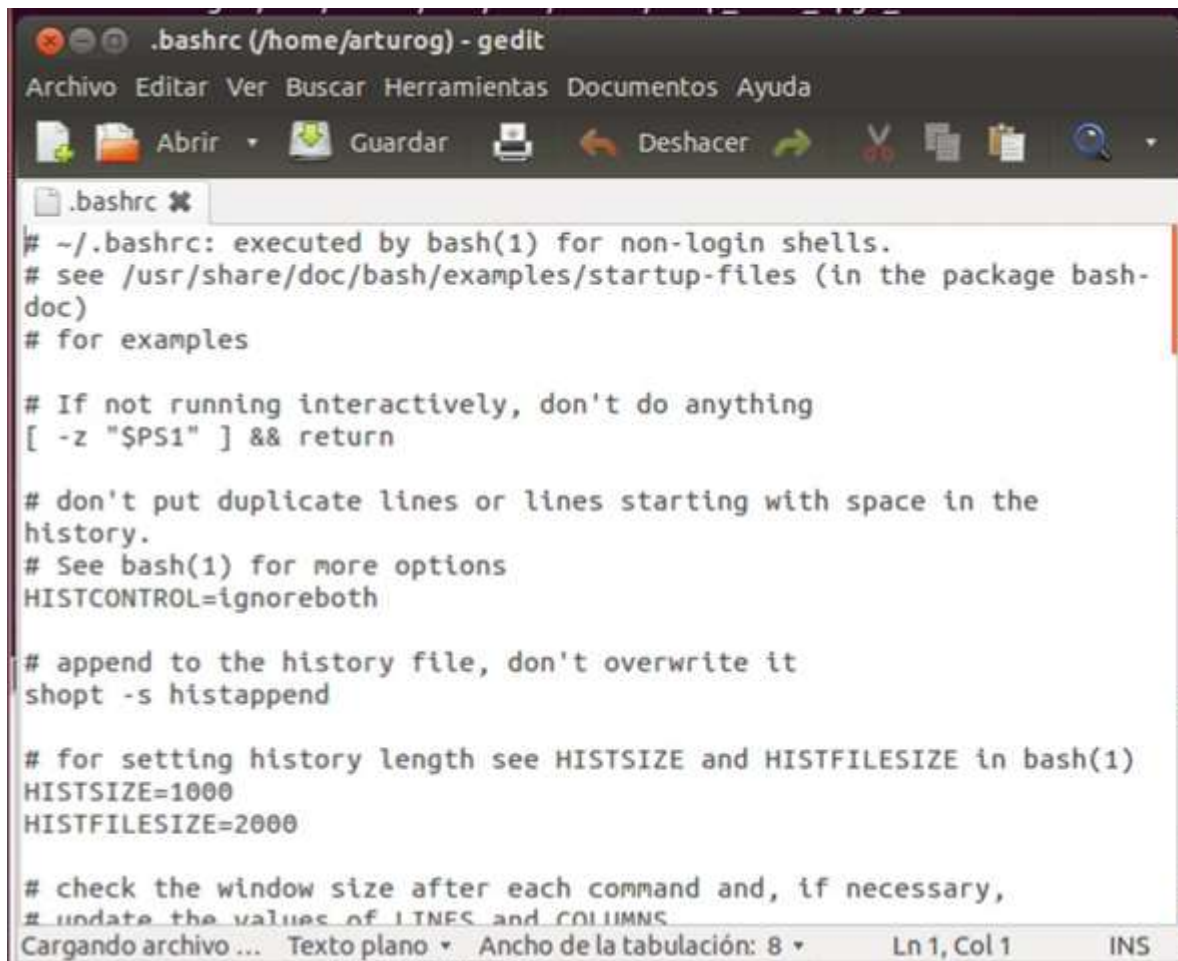
-*ldconfig*

Una vez ejecutados todos los comandos se procede volver a la carpeta home y abrir el archivo bashrc con el fin dar las últimas configuraciones, de la siguiente manera:

-*cd \$HOME*

-*gedit .bashrc*

Al ejecutar, se reflejará algo similar a lo mostrado en la figura 17



```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-
# doc)
# for examples

# If not running interactively, don't do anything
[ -z "$PS1" ] && return

# don't put duplicate lines or lines starting with space in the
# history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS
```

Figura 17 Archivo. bashrc (fuente propia).

Una vez realizado el paso anterior, se terminan de instalar las librerías necesarias agregando las siguientes líneas al final del archivo, y se procede a guardarlo y cerrarlo.

```
-export LD_LIBRARY_PATH=/usr/local/lib
```

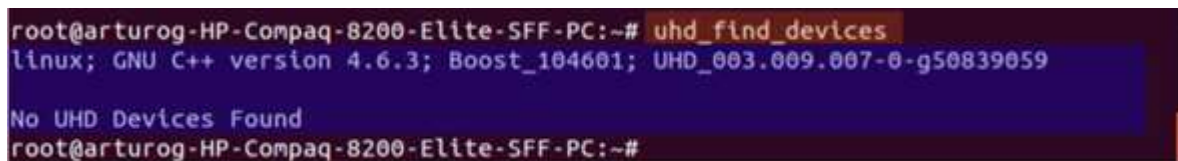
```
-export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

Con el fin que este cambio tenga efecto, se deberá cerrar la ventana del terminal actual y abrir una nueva.

En este punto, los archivos necesarios para ejecutar el UHD debe estar instalados y listos. Esto se puede corroborar, sin un dispositivo USRP conectado, ejecutando el siguiente comando.

```
-uhd_find_devices
```

Se deberá observar en la pantalla una respuesta similar a lo mostrado a la figura 18.

A terminal window screenshot with a dark background. The prompt is 'root@arturog-HP-Compaq-8200-Elite-SFF-PC:~#'. The command 'uhd_find_devices' is entered and highlighted in orange. The output is 'linux; GNU C++ version 4.6.3; Boost_104601; UHD_003.009.007-0-g50839059' followed by 'No UHD Devices Found' on a new line. The prompt is repeated at the bottom.

```
root@arturog-HP-Compaq-8200-Elite-SFF-PC:~# uhd_find_devices
linux; GNU C++ version 4.6.3; Boost_104601; UHD_003.009.007-0-g50839059
No UHD Devices Found
root@arturog-HP-Compaq-8200-Elite-SFF-PC:~#
```

Figura 18 Respuesta del comando `uhd_find_devices` sin la USRP conectada (fuente propia).

La respuesta obtenida en la figura anterior da por finalizada la instalación del paquete UHD en la distribución Ubuntu 12.04.

El siguiente paso a continuación, es realizar la configuración del radio USRP con el paquete OpenBTS. Se abre nuevamente la terminal de comandos y se ingresa en modo súper-usuario.

```
-cd /usr/local/src
```

Una vez allí, se importa el repositorio del OpenBTS y se crea la carpeta `openbts` ejecutando las instrucciones mostradas a continuación en el siguiente orden:

```
- git svn init http://wush.net/svn/range/software/public openbts - git svn init
http://wush.net/svn/range/software/public openbts
```

```
- cd openbts/
```

```
- git svn fetch
```

```
- ./git-svn-clone-externals
```

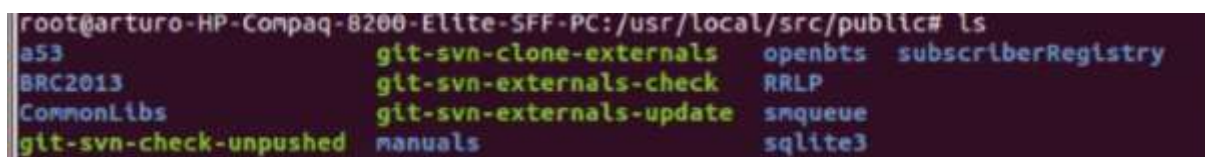
Una vez realizado lo anterior, se se regresa a la carpeta src, y una vez allí, se procede a descargar el código principal de la OpenBTS, ejecutando los siguientes comandos:

- svn co <http://wush.net/svn/range/software/public>

Al terminar el proceso anterior, quedará una carpeta con el nombre public, como se muestra en la figura 19; dentro de esa carpeta se harán las instalaciones de la OpenBTS. Se debe ingresar a la carpeta public y a continuación se entra a la carpeta a3 y se ejecutan los siguientes comandos:

-cd a3/trunk/

-sudo make install.



```
root@arturo-HP-Compaq-8200-Elite-SFF-PC:/usr/local/src/public# ls
a3                               git-svn-clone-externals        openbts  subscriberRegistry
BRC2013                         git-svn-externals-check        RRLP
CommonLibs                     git-svn-externals-update       smqueue
git-svn-check-unpushed         manuals                        sqlite3
```

Figura 19 Visualización de la carpeta public (fuente propia).

Una vez hecho esto, se debe volver a la carpeta public y se ingresa a la carpeta openbts y se crea el soporte para dispositivos UHD siguiendo los siguientes comandos:

-cd openbts/trunk/

-autoreconf -i

./configure

./configure --with-uhd

-make

Otro aspecto necesario es crear un enlace simbólico para el archivo transceiver52. Para esto, dentro de la carpeta openbts se entra a una llamada apps y introducir el siguiente enlace.

```
-ln -s ../Transceiver52M/transceiver
```

Posteriormente se regresa a la carpeta anterior y se procede a crear el “Bloque de base de datos” para todo el OpenBTS, el cual se describe en la figura 11. Para esto se ejecutan los siguientes comandos:

```
-sudo mkdir /etc/OpenBTS
```

```
-sudo sqlite3 -init ./apps/OpenBTS.example.sql /etc/OpenBTS/OpenBTS.db ".quit"
```

```
-sqlite3 /etc/OpenBTS/OpenBTS.db .dump
```

Una próxima acción es crear la ruta para configurar la “Base de datos del registro de suscriptor”, el cual es el bloque que se encuentra en la figura 11; esto se realiza en la carpeta public introduciendo el siguiente comando:

```
-sudo mkdir -p /var/lib/asterisk/sqlite3dir
```

Después de hacer lo anterior, se entra a subscriberRegistry para construir el servidor de autenticación SIP, que se encuentra en el “Bloque de servicio de autenticación y registro” descrito anteriormente; esto se logra ejecutando los siguientes comandos:

```
-cd subscriberRegistry/trunk/
```

```
-make
```

```
-sudo sqlite3 -init subscriberRegistry.example.sql /etc/OpenBTS/sepauthserve.db ".quit"
```

De esta forma ya se arma el bloque de autenticación y registro. Para verificar su instalación, dentro de esta carpeta se ejecuta el comando

```
-./sipauthserve
```

Este debe arrojar la siguiente respuesta:

```
ALERT 140030453958464 15:28:03.6 sipauthserve.cpp:277:main: ./sipauthserve
```

```
(re)starting
```

Esta respuesta nos indica que se cuenta con este servicio instalado.

A continuación, se regresa a la carpeta public y se procede a crear el “Bloque de servicios de mensajería instantánea” que se describe en la figura 11; esta acción se logra entrando a la carpeta smqueue y ejecutando los siguientes comandos.

```
-cd smqueue/trunk/  
-autoreconf -i  
-./configure  
-make  
-sudo sqlite3 -init smqueue/smqueue.example.sql /etc/OpenBTS/smqueue.db ".quit"
```

De esta forma ya está armado el “bloque de servicios de mensajería instantánea”. Para ejecutarlo, dentro de esta carpeta se encuentra una denominada smqueue, se ingresa en ella y se ejecuta:

```
-cd smqueue/  
-./smqueue
```

Esto arrojará el siguiente resultado:

```
ALERT 140061611308864 16:26:10.0 smqueue.cpp:2651: main: smqueue (re)starting  
smqueue logs to syslogd facility LOCAL7, so there's not much to see here
```

Esto indicará que este bloque de servicios de mensajería instantánea, se encuentra instalado. Ya armado el gestor de mensajes, se regresa a la carpeta public y así queda culminado la instalación del OpenBTS.

5.3.2. Configuración y ejecución del OpenBTS en el radio USRP.

Para comunicar la herramienta de radio USRP con el computador se realiza la conectividad a través de un cable de red Gigabit en el puerto Ethernet del N210 al puerto Ethernet del equipo de cómputo. Una vez hecho esto, desde el equipo de cómputo se

selecciona Configuración del sistema – Red – Cableada y en la sección de configuración se selecciona conectar automáticamente, como se muestra en la figura 20.

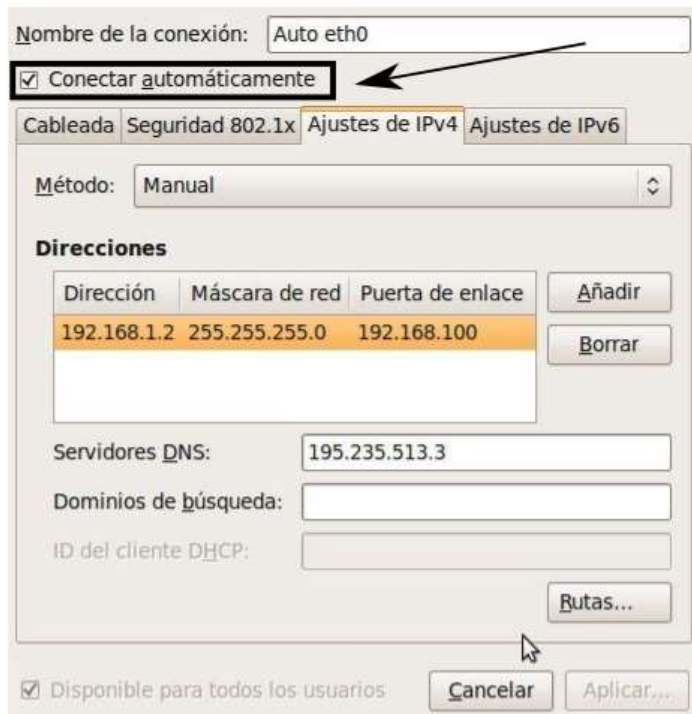


Figura 20 Menú de configuración de red cableada (fuente propia).

Luego, se configura la IP del host para establecer una red que posibilite la comunicación con el radio USRP N210; este viene con una configuración de IP predeterminada establecida en 192.168.10.2. Para esta configuración se accede la terminal y en modo súper-usuario se ejecutan los comandos para configurar el host y se verifica la conectividad USRP y PC como se indica a continuación:

- ifconfig eth0 192.168.10.1
- ping 192.168.10.2

Una vez verificada la conexión se ejecuta el reconocimiento del radio USRP por medio del siguiente comando:

- uhd_find_devices

Una vez se logre reconocer el radio, ya el equipo estará listo para la ejecución del OpenBTS

Para ejecutar el OpenBTS en el radio USRP, se habilitan inicialmente el servidor de autenticación y registros, y el de mensajería instantánea. Para el servidor de autenticación y registros se accede a la terminal en modo súper-usuario y se ejecutan los siguientes comandos:

```
-cd /usr/local/src/public/subscriberRegistry/trunk/  
-./sipauthserve
```

Para el servidor de mensajería instantánea se ejecutan las siguientes instrucciones:

```
-cd /usr/local/src/public/smqueue/trunk/smqueue/  
-./smqueue
```

Posteriormente se activa el transceiver del OpenBTS para verificar posteriormente su funcionamiento, ejecutando las siguientes instrucciones en la terminal en modo súper-usuario:

```
-cd /usr/local/src/public/openbts/trunk/apps/  
-./OpenBTS
```

Finalmente, se configuran los parámetros del OpenBTS, para lo cual en la terminal en modo súper-usuario se ejecutan las siguientes líneas:

```
-cd /usr/local/src/public/openbts/trunk/apps/  
-./OpenBTSCLI
```

Si es la primera vez que se usa el OpenBTS en la ventana de este comando se realiza la siguiente acción:

```
config Control.LUR.OpenRegistration .*
```

Esta acción elimina o libera las restricciones de registro IMSI, para que cualquier con una SIM personalizada o comercial pueda entrar a la red. Con esto ya se tiene OpenBTS con todos sus componentes funcionando.

5.3.3. Configuración de un sistema de conmutación basado en Asterisk para el OpenBTS.

Para poder utilizar la base de datos en tiempo real de la herramienta OpenBTS y conmutar llamadas por medio de Asterisk, se hicieron los siguientes procedimientos utilizando como referencia la guía “OpenBTS: 2.8 Instalasi Real Time Asterisk”. Para esto, se reinstalarán las herramientas Asterisk y sqlite3 de la siguiente forma:

```
-sudo apt-get remove asterisk sqlite3
```

```
-sudo apt-get autoremove
```

A continuación, se descarga sqlite del repositorio:

<http://www.sqlite.org/sqlite-autoconf-3071300.tar.gz>

Una vez descargado, desde la terminal se ubica la carpeta donde se descargó y se ejecutan los siguientes comandos para redireccionar e instalar sqlite.

```
-cp sqlite-autoconf-3071300.tar.gz /usr/local/src/
```

```
-cd /usr/local/src/
```

```
-tar zxvf sqlite-autoconf-3071300.tar.gz
```

```
-cd /usr/local/src/sqlite-autoconf-3071300
```

```
-. ./configure CFLAGS="-DSQLITE_ENABLE_COLUMN_METADATA -O2"
```

```
-make clean
```

```
-make all
```

```
-make install
```

A continuación, se instalarán algunas librerías necesarias sobre Ubuntu.

Tabla 4 *Librerías adicionales para Asterisk*

Comando	Dependencia
sudo apt-get install	libsqlite3-dev
	erlang
	unixodbc
	Erlang
	bison
	unixodbc-dev
	openssl
	libssl-dev
	mysql-server
	perl-modules
	libxml2-dev
	libc6-dev
	libnewt-dev
	zlib1g-dev

Nota. Librerías adicionales para Asterisk, con comandos y dependencias correspondientes. (fuente propia).

Una vez hecho estos pasos se procede a descargar el servicio de Base de datos del OpenBTS sqliteodbc, entrando a siguiente enlace.

<http://www.ch-werner.de/sqliteodbc/sqliteodbc-0.95.tar.gz>

Una vez hecha esta acción, desde la terminal ubicamos la carpeta de descarga y ejecutamos los siguientes comandos para redireccionar e instalar.

```
-cp sqliteodbc-0.95.tar.gz /usr/local/src/
```

```
-cd /usr/local/src
```

```
-tar zxvf sqliteodbc-0.95.tar.gz
```

```
-cd /usr/local/src/sqliteodbc-0.95
```

```
./configure
```

```
-make clean
```

```
-make all
```

-make

-make install

El siguiente paso consiste en reinstalar Asterisk realtime. Para esto se descarga la herramienta usando el siguiente enlace.

<http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-10.5.1.tar.gz>

Una vez descargado desde la terminal se ubica la dirección de carpeta donde se descargó y se realizan las siguientes acciones para redireccionar e instalar.

-cp asterisk-10.5.1.tar.gz /usr/local/src/

-cd /usr/local/src

-tar zxvf asterisk-10.5.1.tar.gz

-cd /usr/local/src/asterisk-10.5.1

./configure --disable-xmldoc

-make menuselect

En este punto durante la instalación se revisa el menú de ajustes de Asterisk, y se verifican las que las siguientes opciones de la figura 21 estén marcadas.

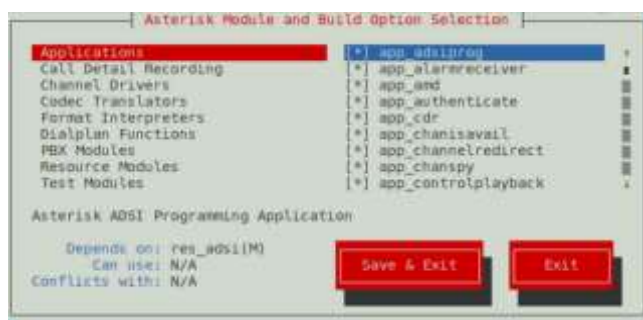


Figura 21 Menú de ajustes instalación Asterisk (fuente propia).

De la figura 21 es importante corroborar que exista una marca con “*” en las siguientes opciones:

- > Dialplan Functions -> func_odbc
- > Utilities -> astdb2sqlite3
- > PBX Modules -> pbx_realtime
- > "Resource Modules" -> "res_realtime"

Además, se debe verificar que no exista "XXX" en las opciones Sqlite3 y ODBC en el menú de configuración. Una vez hecho esto, se procede a guardar los cambios y salir de las opciones.

El proceso de instalación Asterisk se continúa con las siguientes instrucciones:

- make
- make install
- chown -Rf asterisk.asterisk /var/log/asterisk/

Ya reinstalado Asterisk se hacen los siguientes ajustes, haciendo uso de la terminal en el modo súper-usuario, en donde ubicamos la carpeta etc y se ingresa a odbcinet para agregar y editar lo siguiente:

- cd /etc/
- gedit odbcinet.ini
- [SQLite3]
- Description=SQLite3 ODBC Driver
- Driver=/usr/local/lib/libsqlite3odbc.so
- Setup=/usr/local/lib/libsqlite3odbc.so
- Threading=2

Posteriormente, en la misma carpeta etc abrir odbc para agregar y editar lo siguiente.

- gedit odbc.ini

[asterisk]

Description=SQLite3 database

Driver=SQLite3

Database=/var/lib/asterisk/sqlite3dir/sqlite3.db

optional lock timeout in milliseconds

Timeout=2000

Para garantizar la operación de la base de datos se instala el ODBC (Open DataBase Connectivity) y se añade un enlace simbólico. Para esto, haciendo uso de la terminal se siguen las siguientes acciones:

```
-cd /usr/local/etc; ln -s /etc/odbc.ini; ln -s /etc/odbcinst.ini  
-cd /root; ln -s /etc/odbc.ini .odbc.ini; ln -s /etc/odbcinst.ini .odbcinst.ini  
-cd ~asterisk; ln -s /etc/odbc.ini .odbc.ini; ln -s /etc/odbcinst.ini .odbcinst.ini  
-cd ~openbts; ln -s /etc/odbc.ini .odbc.ini; ln -s /etc/odbcinst.ini .odbcinst.ini  
-cp /usr/local/src/openbts/openbts/trunk/AsteriskConfig/*conf /etc/asterisk/
```

Una vez hecho esto, se habilita la configuración de Asterik para utilizar la base de datos ODbC ubicando inicialmente la carpeta de Asterik; dentro de esta carpeta se realizarán modificaciones que permitan usar la base datos en tiempo real por parte de Asterik, tal como se indican en las siguientes instrucciones.

```
-cd /etc/asterisk/  
-gedit modules.conf  
[modules]  
autoload=yes  
; noload => res_config_odbc.so  
preload => res_odbc.so  
preload => res_config_odbc.so
```

```
-gedit extconfig.conf
```

```
[settings]
```

```
sipusers => odbc,asterisk,sip_buddies
```

```
sippeers => odbc,asterisk,sip_buddies
```

```
-gedit res_odbc.conf
```

```
[asterisk]
```

```
enabled => yes
```

```
dsn => asterisk
```

```
pre-connect => yes
```

```
-gedit func_odbc.conf
```

```
[SQL]
```

```
dsn=asterisk
```

```
readsql=${ARG1}
```

```
; ODBC_PRESENCE - Retrieve and update presence
```

```
[PRESENCE]
```

```
dsn=asterisk
```

```
readsql=SELECT location FROM presence WHERE id='${SQL_ESC(${ARG1})}'
```

```
writesql=UPDATE presence SET location='${SQL_ESC(${VAL1})}' WHERE
```

```
id='${SQL_ESC(${ARG1})}'
```

```
-gedit extensions.conf
```

```
[phones]
```

```
; This is the context for handsets provisioned through the realtime database.
```

```
exten => _N.,1,Set(Name=${ODBC_SQL(select dial from dialdata_table where exten =
```

```
\"${EXTEN}\")})
```

```
exten => _N.,n,GotoIf(["${Name}" = ""] ?outbound-trunk,${EXTEN},1)
```

```
exten => _N.,n,Set(IPAddr=${ODBC_SQL(select ipaddr from sip_buddies where name =
\"${Name}\")})
```

```
exten => _N.,n,GotoIf(["${IPAddr}" = "" ] ?outbound-trunk,${EXTEN},1)
```

```
exten => _N.,n,Dial(SIP/${Name}@${IPAddr}:5062)
```

Una vez terminado el proceso anterior ejecutar lo siguiente.

```
-tail -f /var/log/asterisk/full
```

```
-core restart gracefully
```

```
-sudo chown -R asterisk:asterisk /var/lib/asterisk/sqlite3dir
```

Después en la carpeta asterisk se realiza la configuración que permite el registro de un usuario nuevo, realizándose las siguientes acciones

```
-gedit sip.conf
```

```
[IMSIXXXXXXXXXXXXXXXXXX]
```

```
callerid=ZZZZZZZ
```

```
canreinvite=no
```

```
type=friend
```

```
allow=gsm
```

```
context=sip-external
```

```
host=dynamic
```

```
dtmfmode=info
```

```
-gedit extentions.conf
```

```
[macro-dialGSM]
```

```
exten => s,1,Dial(SIP/${ARG1},20)
```

```
exten => s,2,Goto(s-${DIALSTATUS},1)
```

```
exten => s-CANCEL,1,Hangup
```

```
exten => s-NOANSWER,1,Hangup
```



```
exten => s-BUSY,1,Busy(30)
```

```
exten => s-CONGESTION,1,Congestion(30)
```

```
exten => s-CHANUNAVAIL,1,playback(ss-noservice)
```

```
exten => s-CANCEL,1,Hangup
```

```
[sip-external]
```

```
Exten=>ZZZZZZZ,1,Macro(dialGSM,IMSIXXXXXXXXXXXXXXXXXX@127.0.0.1:5062).
```

Para hacer el registro de un nuevo usuario, se registran el número de este en la opción IMSI (International Mobile Subscriber Identity) y el número de teléfono que se desea asignar (CalleID). Para esto se ejecuta la OPenBTS y se abre una ventana Asterisk en modo súper-usuario.

```
-asterisk -rvvvvvv
```

Una vez el equipo terminal móvil se conecte a la red mostrada por la OpenBTS, se utiliza el comando tmsis en el terminal de openbtsccli con el fin de ver su número IMSI; una vez hecho esto, desde la carpeta Asterisk, se agregan los datos de IMSI y CallerID en los siguientes archivos, donde Z representa CallerID y X representa IMSI

```
sip.conf
```

```
[IMSIXXXXXXXXXXXXXXXXXX]
```

```
callerid=ZZZZZZZ
```

```
canreinvite=no
```

```
type=friend
```

```
allow=gsm
```

```
context=sip-external
```

```
host=dynamic
```

```
dtmfmode=info
```

```
extentions.conf
```

Exten=>ZZZZZZZ,1,Macro(dialGSM,IMSIXXXXXXXXXXXXXXXXXX@127.0.0.1:5062).

Este procedimiento se hará por cada terminal móvil que se conecte y se quiera habilitar para realizar llamadas. Una vez terminado el registro del equipo se va a la terminal donde se ejecuta Asterisk, y se realizar las siguientes acciones para actualizar el registro en el programa.

-sip reload

-extensions reload

-dialplan reload

-reload

Estas acciones descritas anteriormente, permitirán la operación de Asterisk con la OPenBTS sin complicaciones; de necesitarse a requerir un reinicio, se ejecutará el siguiente comando.

-core restart gracefully

Estas acciones garantizan la posibilidad de hacer las llamadas con los equipos registrados.

5.4. Pruebas de conectividad de la herramienta OpenBTS entre usuarios móviles a través del envío de mensajes y llamadas de voz GSM.

En esta sección se llevaron a cabo diferentes pruebas para corroborar el funcionamiento de la herramienta OpenBTS y determinar algunas de sus limitantes, enfocadas a número de usuarios conectados al tiempo, llamadas simultáneas y rango de operación de la herramienta en diferentes escenarios.

5.4.1. Pruebas para medir la capacidad de llamadas y mensajes según el número de equipos conectados

En esta prueba se busca evaluar los mensajes de texto y llamadas entre usuarios conectados. Para esto realizaron tests de conectividad con múltiples teléfonos celulares, de diez modelos distintos, aplicándose la siguiente metodología para la realización de los tests:

- Conectar un dispositivo y realizar la comunicación desde el OpenBTS instalado en el equipo de cómputo
- Realizar la conectividad entre dos dispositivos y verificar su comunicación.
- Incrementar el número de dispositivos conectados a la OpenBTS, de dos en dos, hasta llegar a los 10 usuarios.

Dentro de los equipos utilizados, se tienen los mostrados en la tabla 5:

Tabla 5 *Equipos compatibles y usados en la prueba de conectividad de usuarios*

Equipos usados por la herramienta OpenBTS
Moto G5
IPHONE 5
HUAWEI MATE 20
LG K40
XIAOMI REDMI 6 ^a
MOTOROLA ONE VISION
XIAOMI NOTE 7

XIAOMI REDMI NOTE 4

XIAOMI MI A2 LITE

XIAOMI NOTE 9

Nota. Equipos compatibles y usados en la prueba de conectividad de usuarios (fuente propia).

Inicialmente los equipos se conectaron a la red identificada por la herramienta OpenBTS. Esto era notorio, ya que el terminal móvil envía un mensaje de texto desde el usuario con ID 101, el cual permite identificar el número a registrar; en algunos equipos este proceso no se realizaba, en tal situación, se envió un mensaje con el número del usuario a registrar al ID 101; posteriormente, se esperó la respuesta con ID del móvil, lo que indicaba el registro del usuario en la red GSM de la herramienta OpenBTS.

Posteriormente los equipos conectados realizaron llamadas y envío de mensajes de textos entre ellos. Los resultados de estas pruebas son analizados en el capítulo 6. Una imagen de las llamadas realizadas se muestra en la figura 22.



Figura 22 Equipos con llamadas en curso usando herramienta OpenBTS (fuente propia).

5.4.2. Pruebas de consumo de recursos red GSM.

Para esta prueba se realizó una captura de información al realizar llamadas entre dos usuarios en diferentes escenarios para los tests; posteriormente se analizó el consumo promedio en la comunicación, en cuanto a recursos como procesador, memoria RAM, uso de red en el envío y recepción de información en (MiB/s). Para esta prueba se ubicó la herramienta OpenBTS en la plazoleta que se encuentra entre los bloques 1, 2 y 3 de la Universidad de la Costa, y se procedió con las capturas de consumo de recursos. Esto se muestra en la figura 24.



Figura 23 Ubicación de la herramienta OpenBTS para la prueba 2 (fuente propia).

Esta segunda prueba analiza el uso de recursos de la OpenBTS, utilizando como herramienta el monitor del sistema de Ubuntu, con el fin de tomar muestras relacionadas al consumo de memoria y capacidad de procesamiento en el computador al realizar llamadas y envío de mensajes.

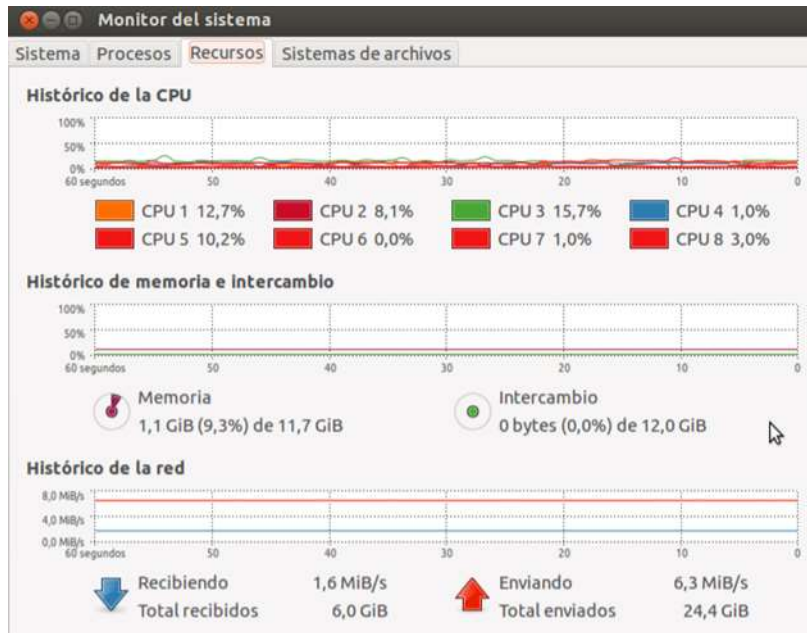


Figura 24 Monitor de sistema de Ubuntu que muestra consumo de recursos al usar OpenBTS (fuente propia).

Los datos que se obtendrán serán los siguientes: procesador (%), memoria RAM (%), uso de red en el envío y recepción de información en (MIB/s). Se compararon 7 muestras con llamadas de usuarios para evaluar el consumo de la OpenBTS en estos parámetros, y se compararon con el consumo del equipo de cómputo sin esta herramienta. Los datos obtenidos se observan en las tablas 6 y 7.

Tabla 6 *Datos de transferencias de datos*

Procesador (%)	Memoria (11.7) (%)	Red enviada (MIB/s)	Red recibida (MIB/s)
2,1	8,9	0	0
6,4125	9	0	0
6,425	8,9	0	0
5,625	8,9	0	0
10,975	9	0	0
16,0375	8,9	0	0
16,0375	8,9	0	0
13,1	9	0	0

Nota. Consumo del equipo cómputo con la herramienta OpenBTS activa con llamada (fuente propia).

Tabla 7 *Datos de procesamiento de consumo del computador*

Procesador (%)	Memoria (11.7) (%)	Red enviada (MIB/s)	Red recibida (MIB/s)
8,4125	9.2	1.6	6.3
6,4625	9.3	1.6	6.3
4,9875	9.1	1.6	6.3
9,275	9.1	1.6	6.3
5,6625	9.1	1.6	6.3
9,9875	9.1	1.6	6.3
6,6	9.1	1.6	6.3
5,7125	9.1	1.6	6.3

Nota. Consumo del computador sin la herramienta OpenBTS activa (fuente propia).

5.4.3. Pruebas de calidad de llamada y mensajes SMS según la distancia cubierta.

En esta tercera prueba se busca analizar el rendimiento de la herramienta en un escenario de aplicación real de la herramienta OpenBTS, teniendo en cuenta factores tales como la distancia en cuanto a niveles de recepción de señal del móvil para su uso dentro de la Universidad de la Costa. Para esto se analizará la pérdida de calidad de señal que se aprecia en las llamadas al alejarse de la radiobase (Herramienta OpenBTS). Para esta prueba se definió como escenario de aplicación, las instalaciones de la Universidad de la Costa, en Barranquilla, donde se realizaron varios test de llamadas y envío de mensajes de texto; estos se analizaron en tiempo real, y se obtuvieron las variaciones en la calidad de la señal en el receptor; para obtener esta información se usó el comando `chans -I` en el terminal donde se

ejecuta OpenBTSCLI (OpenBTS), obteniéndose datos de DNLEV que indican el nivel de RSSI que el teléfono móvil receptor informa y que son expresados en dBm.

Para minimizar errores en cuanto los datos obtenidos, en los resultados, se tomaron cinco (5) muestras por cada distancia propuesta, tanto para las llamadas como para los mensajes, esto es, por cada distancia, se repitió la prueba cinco veces, y se promediaron los resultados.

Además de esto se plantearon tres despliegues dentro del escenario de aplicación, que fueron test a cero metros de altura, a 7,2 metros de altura (piso 3 del bloque de Posgrados de la Universidad de la Costa) y a 16,8 metros de altura (Piso 7 del bloque de posgrados de la Universidad de la Costa); esto se realizó para evaluar el rendimiento de la herramienta OpenBTS en posibles situaciones de uso dentro de la Universidad de la Costa, teniendo en cuenta que es un escenario de propagación Urbano, con obstáculos y circulación de personas y vehículos. Esto muestra una zona para las pruebas que ejemplifica una situación real de aplicación.

Con base en lo anteriormente definido, se tienen las siguientes pruebas de aplicación para esta prueba:

- Test 1: Se ubicó la herramienta OpenBTS a 7,2 metros de altura, en el piso 3 del bloque 11 (edificio de posgrados) de la Universidad de la Costa, como se observa en la figura 25. Se realizaron pruebas de llamadas y envíos de mensajes entre dos usuarios, uno cercano al dispositivo y otro separado a distancias de 0 - 5 - 10 - 20 - 40 - 60 - 80 - 100 metros respectivamente; para cada distancia se realizaron 5 llamadas y envíos de mensajes, para minimizar el error.
- Test 2: Se ubicó la herramienta OpenBTS a 16,8 metros de altura, en el piso 7 del bloque 11 (edificio de posgrados) de la Universidad de la Costa, como se observa en la

figura 25. Se realizaron pruebas de llamadas y envíos de mensajes entre dos usuarios, uno cercano al dispositivo y otro separado a distancias de 0 - 5 - 10 - 20 - 40 - 60 - 80 - 100 metros respectivamente; para cada distancia se realizaron 5 llamadas y envíos de mensajes, para minimizar el error.



Figura 25 Ubicación de la herramienta OpenBTS para los test 1 y 2 (fuente propia).

Una vez ejecutados las pruebas se obtuvieron los siguientes datos en cada una de las pruebas, los cuales están consignados en las tablas 8 y 9

Tabla 8 *Mediciones de nivel de señal*

Dist acia	Muestra 1 - Nivel de señal (dBm)	Muestra 2 - Nivel de señal (dBm)	Muestra 3 - Nivel de señal (dBm)	Muestra 4 - Nivel de señal (dBm)	Muestra 5 - Nivel de señal (dBm))	Nivel de señal promedio (dBm)
0	-58	-58	-66	-70	-63	-63
5	-63	-63	-66	-78	-74	-68,8
20	-62	-77	-78	-76	-78	-74,2

40	-71	-78	-72	-79	-75	-75
60	-76	-82	-76	-74	-75	-76,6
80	-94	-101	-94	-95	-77	-92,2
100	-91	-83	-81	-69	-71	-79

Nota. Mediciones de nivel de señal obtenidas en el equipo móvil receptor en el piso 3 del bloque 11 de la Universidad de la Costa (fuente propia).

Tabla 9 *Mediciones de nivel de señal*

Distancia	Muestra 1 - Nivel de señal (dBm)	Muestra 2 - Nivel de señal (dBm)	Muestra 3 - Nivel de señal (dBm)	Muestra 4 - Nivel de señal (dBm)	Muestra 5 - Nivel de señal (dBm)	Nivel de señal promedio (dBm)
0	-95	-79	-79	-86	-85	-84,8
5	-81	-80	-77	-74	-80	-78,4
10	-73	-76	-76	-77	-76	-75,6
20	-79	-76	-79	-79	-85	-79,6
40	-82	-75	-78	-83	-78	-79,2
60	-82	-86	-76	-81	-76	-80,2
80	-88	-80	-84	-84	-81	-83,4
100	-92	-106	-99	-99	-93	-97,8

Nota. Mediciones de nivel de señal obtenidas en el equipo móvil receptor en el piso 7 del bloque 11 de la Universidad de la Costa (fuente propia).

El análisis de esta información se realiza en el siguiente capítulo, a través de gráficas.

6. Resultados y Análisis

Con el fin de evaluar el funcionamiento de la herramienta implementada en este trabajo, se obtuvieron resultados de las pruebas descritas en el capítulo anterior. Estas pruebas buscan determinar y caracterizar la correcta operación de la herramienta, con el fin de identificar los siguientes aspectos:

1. Capacidad de llamadas y mensajes según el número de equipos conectados.
2. Consumo de recursos red GSM.
3. Calidad de llamadas y mensajes SMS según la distancia cubierta.

6.1. Análisis de la capacidad de llamadas y mensajes según el número de equipos conectados.

En la primera prueba se mide la capacidad de ejecución de mensajes y llamadas en la red de acuerdo con el número de celulares conectados. Para tal prueba se usaron varios modelos de celular, conectados y configurados respectivamente en red GSM de OpenBTS, como los mostrados en la tabla 5.

Un aspecto para destacar en esta sección fue que al conectar los equipos el identificador de la red GSM no se percibía de manera similar en los equipos móviles conectados; en algunos terminales la red se percibía con identificadores tales como “001 01” y “TEST PLMN 1 – 1 2G”. Otros equipos mostraban identificadores de la red poco comunes, a los mencionados anteriormente.

Teniendo en cuenta la Tabla 5 los equipos que son compatibles independientemente del modelo son aquellos que no tienen las bandas abiertas y modificaciones de seguridad en su software que lo hace incompatibles con otros operadores o redes desconocidas.

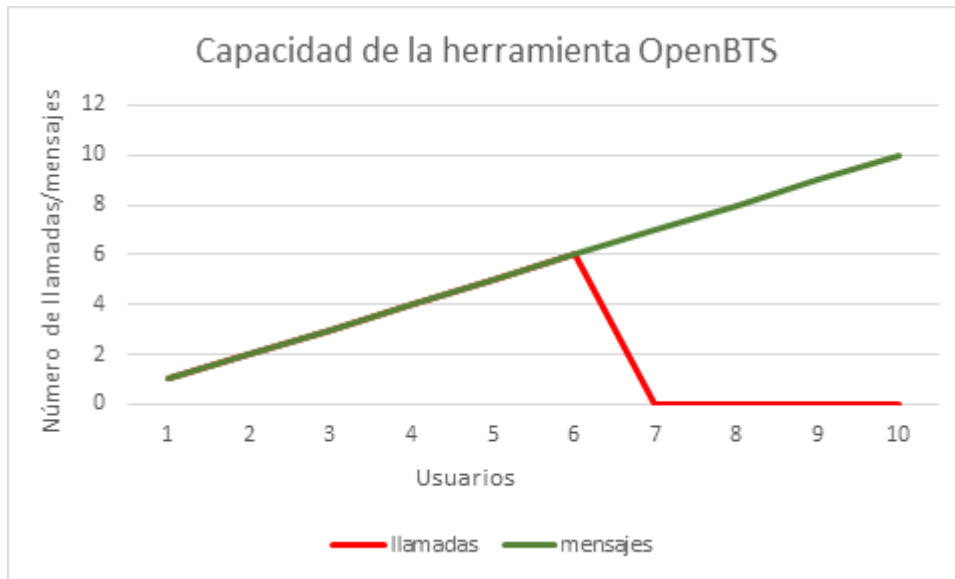


Figura 26 Gráficos de llamadas y mensajes en simultáneo (fuente propia).

De acuerdo con la figura 26 se puede deducir que la capacidad de llamadas se limita a seis (6) equipos terminales, mientras que el servicio de envío de mensaje se mantiene activo en todos los equipos de la prueba; esto quiere decir que de diez (10) equipos terminales probados todos reciben el servicio de mensajería sin problema, mientras que el servicio de llamadas sólo podrá ser usado por seis (6) equipos terminales; los demás equipos podrán usar la OpenBTS en llamadas cuando alguno de los seis anteriores desconecte la llamada. Esto se debe a que la herramienta tiene límites en cuanto al número de llamadas a realizar, y según su documentación está restringido a un máximo de siete (7) simultáneas sin que colapse el sistema. Esto es ligado a la versión usada de la OpenBTS en las pruebas.

6.2. Análisis del consumo de recursos por parte de la herramienta OpenBTS

En esta sección se analizan los datos obtenidos en las tablas 6 y 7, en donde se tomaron siete muestras con la herramienta OpenBTS en funcionamiento y sin esté operando, con el fin de analizar el consumo de recursos sobre un equipo de cómputo y determinar si este era un factor relevante cuando se generan llamadas y se hace uso del equipo. Con base en lo anterior, se promediaron los datos obtenidos en las tablas 6 y 7 y se realizaron las siguientes gráficas que son mostradas en la figura 27.



Figura 27 Comparativa de consumo de recursos ejecutando la herramienta OpenBTS y sin utilizar este equipo (Fuente propia).

Analizando la Figura anterior se puede afirmar que el consumo de recursos de procesamiento, de acuerdo con los datos obtenidos, es mínimo teniéndose que al usar la herramienta OpenBTS no se refleja un aumento en el consumo de recursos de procesamiento en promedio. Para el caso del consumo de recursos de memoria RAM, este presenta un comportamiento similar al del procesador, considerándose despreciable, con 0,2% de aumento al usar la herramienta OpenBTS. En relación con el consumo de recursos de red, la diferencia notoria, esto es debido que solo al conectar el radio USRP fue que se utilizaron recursos de red ethernet; sin embargo, el consumo fue muy bajo, pues fue solamente del orden del 6% al transmitir y del 2% al recibir información.

Con base en el anterior análisis se puede afirmar que el uso de la herramienta OpenBTS no afecta el rendimiento del equipo de cómputo, siendo este de mucha fiabilidad para el uso de esta aplicación.

6.2.1. Calidad de llamadas y mensajes SMS según la distancia cubierta.

Para estas pruebas se analizaron los datos obtenidos en las tablas 8 y 9 del presente documento. Previamente en las pruebas se pudo determinar que un rango óptimo de operación para los test 1 y 2 es de 0 a 100 metros; en este rango se realizaron llamadas y envío de mensajes con circulación de personas y automóviles circulando, y sin edificios que obstaculizaran la señal generada desde la herramienta OpenBTS.

En cuanto a la prueba los resultados obtenidos se analizan en la figura 28, en donde se muestran los niveles promedio de RSSI obtenidos en el receptor a diferentes distancias.

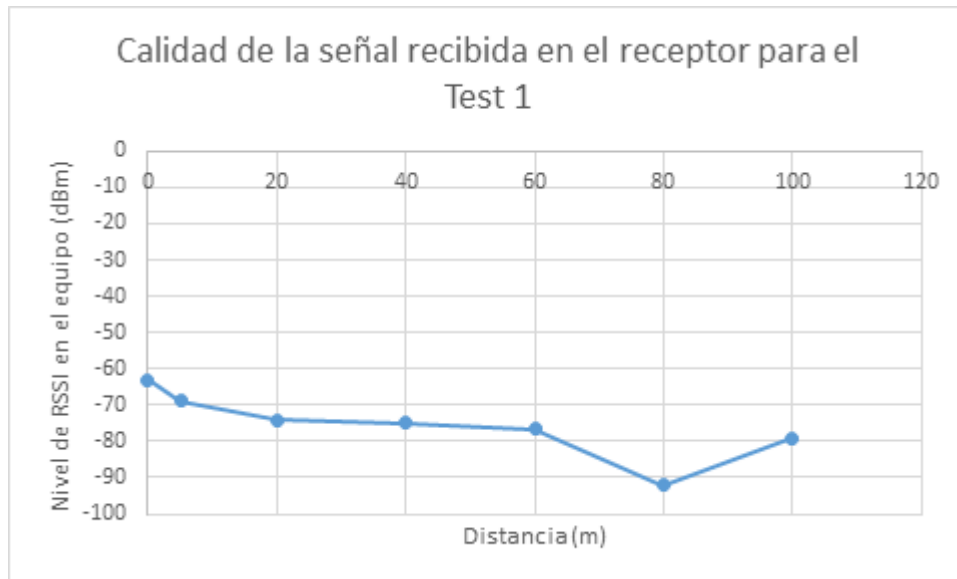


Figura 28 Gráfica de niveles de RSSI obtenidos en el equipo receptor con la herramienta OpenBTS ubicada en el piso 3 del bloque 11 de la Universidad de la Costa Nivel de RSSI (dBm) vs Distancia (m) (Fuente propia).

De la figura 28 se puede determinar que los niveles de RSSI en el receptor disminuyen a medida que la distancia aumenta, lo cual es coherente en la propagación de una señal inalámbrica. Se tiene que el nivel de señal más bajo se obtuvo a una distancia de 80 m, siendo de -92,2 dBm; esto pudo ser causado debido a la circulación de vehículos en alguna de las cinco réplicas realizadas a esta distancia. Los mensajes y las llamadas fueron exitosos en cada punto, aunque las llamadas empezaron a presentar baja calidad a una distancia de 80 metros, una vez que el usuario estaba en movimiento; la calidad de la llamada se estabilizaba una vez el usuario se mantenía en la misma posición.

De manera similar, en cuanto a la prueba 2, los resultados obtenidos se analizan en la figura 29, en donde se muestran los niveles promedio de RSSI obtenidos en el receptor móvil a diferentes distancias.

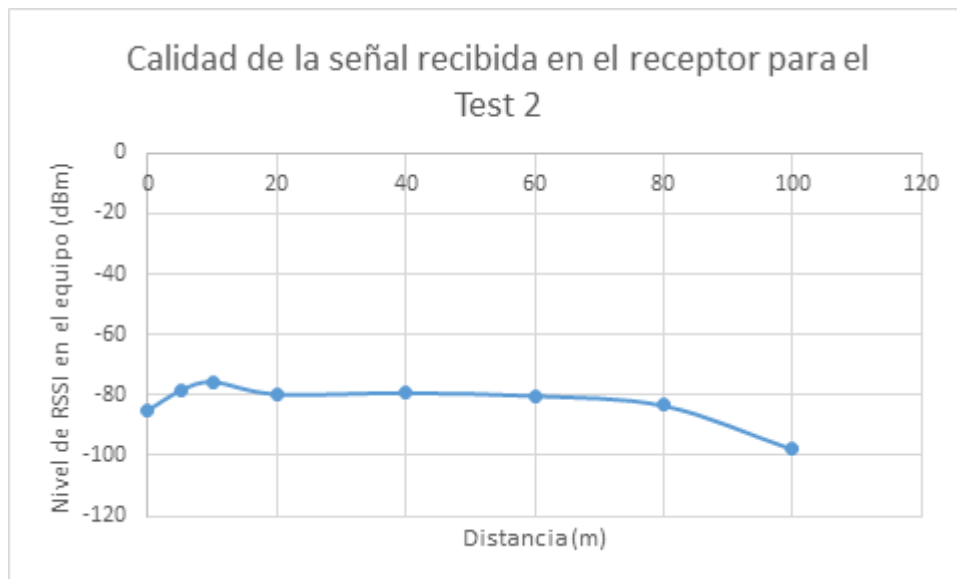


Figura 29 Gráfica de niveles de RSSI obtenidos en el equipo receptor con la herramienta OpenBTS ubicada en el piso 7 del bloque 11 de la Universidad de la Costa Nivel de RSSI (dBm) vs Distancia (m) (Fuente propia).

La figura 29 muestra que en general el comportamiento de la calidad de la señal obtenida en el receptor móvil disminuye a medida que se aleja de la herramienta OpenBTS; esto también es coherente con la propagación de una señal inalámbrica. El valor de menor calidad de la señal en el receptor se obtiene a los 112 m, con -90 dBm. A diferencia del test 1, se obtiene más cobertura, ofreciendo 2 a 3 metros más de la distancia máxima. Al sobrepasar los 112 metros el DNLEV supera los -90 dbm; con estos valores, los mensajes que se transmitían desde el equipo transmisor, ubicado cerca a la herramienta OpenBTS, no llegaban al receptor; así mismo, las llamadas presentaban una calidad deficiente y en algunos casos no se establecían o se desconectaban al corto tiempo; el índice de éxito de una llamada era el promedio de 1 sobre 5.

De los test1 y 2 se evidencia que se tienen mejores resultados en cuanto a nivel de señal recibida, en el primero. Sin embargo, como se mencionó anteriormente, el test 2 permitió obtener una mayor distancia de cobertura.

De esta manera, se finalizaron las pruebas para analizar el funcionamiento de la herramienta OpenBTS, obteniéndose que su funcionamiento fue óptimo y se puede implementar como un equipo de laboratorio para futuras prácticas en asignaturas de posgrado y pregrado.

7. Conclusiones

Se desarrolló una herramienta que emula un sistema de telefonía celular móvil, en donde se obtuvieron un máximo de 10 usuarios conectados y enviando mensajes entre ellos, sin embargo, es posible conectar más dispositivos. Así mismo seis usuarios pudieron realizar llamadas de manera simultánea entre ellos. Se obtuvo un rango máximo de 120 metros en espacio libre, a una altura de 19 metros, también se obtuvo este mismo rango a 9 metros, esto demuestra un excelente rango de cobertura con una calidad óptima. Se realizaron pruebas de consumo de recursos de cómputo del dispositivo, de los cuales podemos concluir que son mínimos, casi irrelevantes en el computador usado lo que nos permitiría usar un equipo con especificación de gama baja o diseñar un circuito embebido para cumplir con los requerimientos de cómputo.

Al realizar el análisis de los resultados obtenidos en el presente proyecto de grado, respecto a la herramienta que emula un sistema de telefonía celular móvil, también se concluye que es una herramienta útil y funcional para el uso educativo, ya emula de forma correcta un sistema tradicional GSM. Sin embargo, no sería viable para el uso comercial por sus diversas limitantes, para este fin se necesitaría un dispositivo de SDR con más capacidad.

7.1 Trabajos futuros

Este proyecto brinda una herramienta con la que se pueden desarrollar muchos proyectos a futuro, a merced del interés de los investigadores, hemos creado unas propuestas de posibles proyectos que se pueden realizar con o en base a la herramienta OpenBTS mostrada en este trabajo:

- Crear guías para realizar prácticas de laboratorio, en asignaturas de telefonía móvil.
- Realizar estudios de campo para estudiar la propagación de la señal.

- Crear aplicaciones de la herramienta para soluciones de transmisión de datos.
- Usar herramientas de desarrollo similares a raspberry pi y sistemas operativos más actuales en versiones de Ubuntu, con el fin de optimizar la herramienta.
- Modificar la herramienta para usar un estándar diferente al 2G

8. Referencias

- A. Gupta, R. J. (28 de julio de 2015). A Survey of 5G Network: Architecture and Emerging Technologies. IEEE, 27.
- Alcocer Erazo, J. D. (2019). Análisis de desempeño del estándar IEEE 802.11AF en TVWS mediante la implementación en una plataforma de radio definido por software. 100.
- Carlos. (10 de 01 de 2014). Obtenido de <https://www.discoduroderoer.es/ubuntu-desktop-12-04-y-12-10-lts/>
- Daniel. (2 de 08 de 2016). www.vozelia.com. Obtenido de <https://www.vozelia.com/es/voz-ip/>
- EcuRed contributors. (4 de 3 de 2014). ecured. Obtenido de https://www.ecured.cu/Radios_definidos_por_Software
- Eqeng. (09 de 05 de 2010). Obtenido de <http://www.eqeng.com/site/es/node/38>
- Ettus. (2010). ettus. Obtenido de https://www.ettus.com/wp-content/uploads/2019/01/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf
- Ettus. (07 de 02 de 2020). ettus. Obtenido de <https://kb.ettus.com/UHD>
- Garzón Gaona, J. G. (2018). Implementación de un receptor ONE SEG utilizando la plataforma de radio definido por software USRP para el estándar ISDB-Tb. Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE, 106.
- Glosarios.servidor-alicante. (14 de 10 de 2016). glosarios.servidor-alicante. Obtenido de <https://glosarios.servidor-alicante.com/telecomunicaciones/estacion-movil>

- GOODIER, R. (02 de 10 de 2015). engineeringforchange. Obtenido de <https://www.engineeringforchange.org/news/open-source-cell-network-could-cut-costs/>
- Hafidudin, M. F. (2019). Implementation of USRP (Universal Software Peripheral Radio) as. iopscience, 12.
- Jianchao Zheng, J. Y. (2018). Modeling and Analysis of Two-tier MIMO Heterogeneous Cellular Network. ScienceDirect, 44.
- Jonathan D. Lehnert, A. S. (2018). Development and pilot testing of a text message vaccine reminder system for use during an influenza pandemic. TAYLOR & FRANCIS, 8.
- Jorge E. Pulgarín, E. P. (2013). GSM system implementation for academic purposes. IEEE, 6.
- Kinjal Aggrawal, K. V. (2017). Reconfigurable Cellular GSM Network using USRP. IEEE, 6.
- Manuel Quiñones, K. R. (03 de 2015). researchgate. Obtenido de https://www.researchgate.net/publication/286439561_Sistema_de_conmutacion_y_control_de_llamadas_basado_en_OpenBTS_y_Asterisk
- Muñoz, W. Y., Orozco, H. F., & Ramos, J. A. (2018). Red de comunicaciones móviles. espacios , 13.
- National Instruments . (2020). ettus. Obtenido de <https://www.ettus.com/all-products/WBX/>
- National Instruments. (01 de 01 de 2019). National Instruments. Obtenido de <https://www.ettus.com/all-products/un210-kit/>
- Openbts.org. (25 de 1 de 2012). <http://openbts.org/>. Obtenido de <http://openbts.org/>

Prasannan, N., Xavier, G., Manikkoth, A., Gandhiraj, R., Peter, R., & Soman, K. P. (2013).

OpenBTS Based Microtelecom Model:A SocioEconomic Boon To Rural

Communities . IEEE, 6.

Guarea. (6 de 4 de 2015). Obtenido de [https://www.quarea.com/es/asterisk-funcionalidades-](https://www.quarea.com/es/asterisk-funcionalidades-basicas-avanzadas)

[basicas-avanzadas](https://www.quarea.com/es/asterisk-funcionalidades-basicas-avanzadas)

Repositorio.unicauca.edu.co. (5 de 9 de 2015). repositorio.unicauca.edu.co. Obtenido de

<http://repositorio.unicauca.edu.co:8080/xmlui/bitstream/handle/123456789/1503/PRO>

[TOTIPO%20DE%20UNA%20CENTRAL%20DE%20TELEFON%c3%8dA%20M%](http://repositorio.unicauca.edu.co:8080/xmlui/bitstream/handle/123456789/1503/PRO)

[c3%93VIL.pdf?sequence=1&isAllowed=y](http://repositorio.unicauca.edu.co:8080/xmlui/bitstream/handle/123456789/1503/PRO)

Roberto, L. G. (26 de 04 de 2015). Obtenido de

[https://riunet.upv.es/bitstream/handle/10251/19023/Memoria_Lizon_Gonzalez_Rober](https://riunet.upv.es/bitstream/handle/10251/19023/Memoria_Lizon_Gonzalez_Roberto.pdf)

[to.pdf](https://riunet.upv.es/bitstream/handle/10251/19023/Memoria_Lizon_Gonzalez_Roberto.pdf)

Shujaat Ali Khan Tanoli, M. R. (2018). An Experimental Channel Capacity Analysis of.

MDPI, 13.

Ubuntu-co. (25 de 03 de 2018). Obtenido de <https://ubuntu-co.com/>

Universidad internacional de valencia . (10 de 10 de 2018). universidadviu. Obtenido de

<https://www.universidadviu.com/que-es-gsm-y-como-funciona/>

Universidadevigo. (15 de 12 de 2018). youtube. Obtenido de Iniciación a SDR (Radio

Definida por Software) y su aplicación en la industria:

<https://www.youtube.com/watch?v=9XZpvNen2t4>

Wikiasterisk. (06 de 07 de 2014). wikiasterisk. Obtenido de

https://www.wikiasterisk.com/index.php/Asterisk_Realtime

Xuesong Cai, A. G.-P. (2017). Low Altitude UAV Propagation Channel Modelling. IEEE, 5.

Varun Nambiar, E. V. (2016). SDR Based Indoor Localization Using Ambient. IEEE, 6.

Zakaria Er-reguig, H. A. (2019). Towards Designing a Microcell Base Station using. IEEE, 4.

Wireshark. (2020). <https://www.wireshark.org/>. Obtenido de <https://www.wireshark.org/>

3cx. (2020). 3cx. Obtenido de https://www.3cx.es/centralita-telefonica/descargar-asterisk/?src=asteriskeslatamads&gclid=Cj0KCQiAqY3zBRDQARIsAJeCVxOdV0FEz29GbvLvmV_ttbz5f_8IrRPOyZ8oyetyGbICVGY8yD7QaqEaAjNeEALw_wcB